VIRGINIO SALA
GUIDA AL
PERSONAL
COMPUTER

VIRGINIO SALA

guida al
personal computer

con oltre 200 illustrazioni a colori e in nero

Sommario

7 A modo di introduzione

15 Guidare il personal computer

37 Il software: ovvero che cosa si può fare

71 Come si fa a farlo: l'hardware

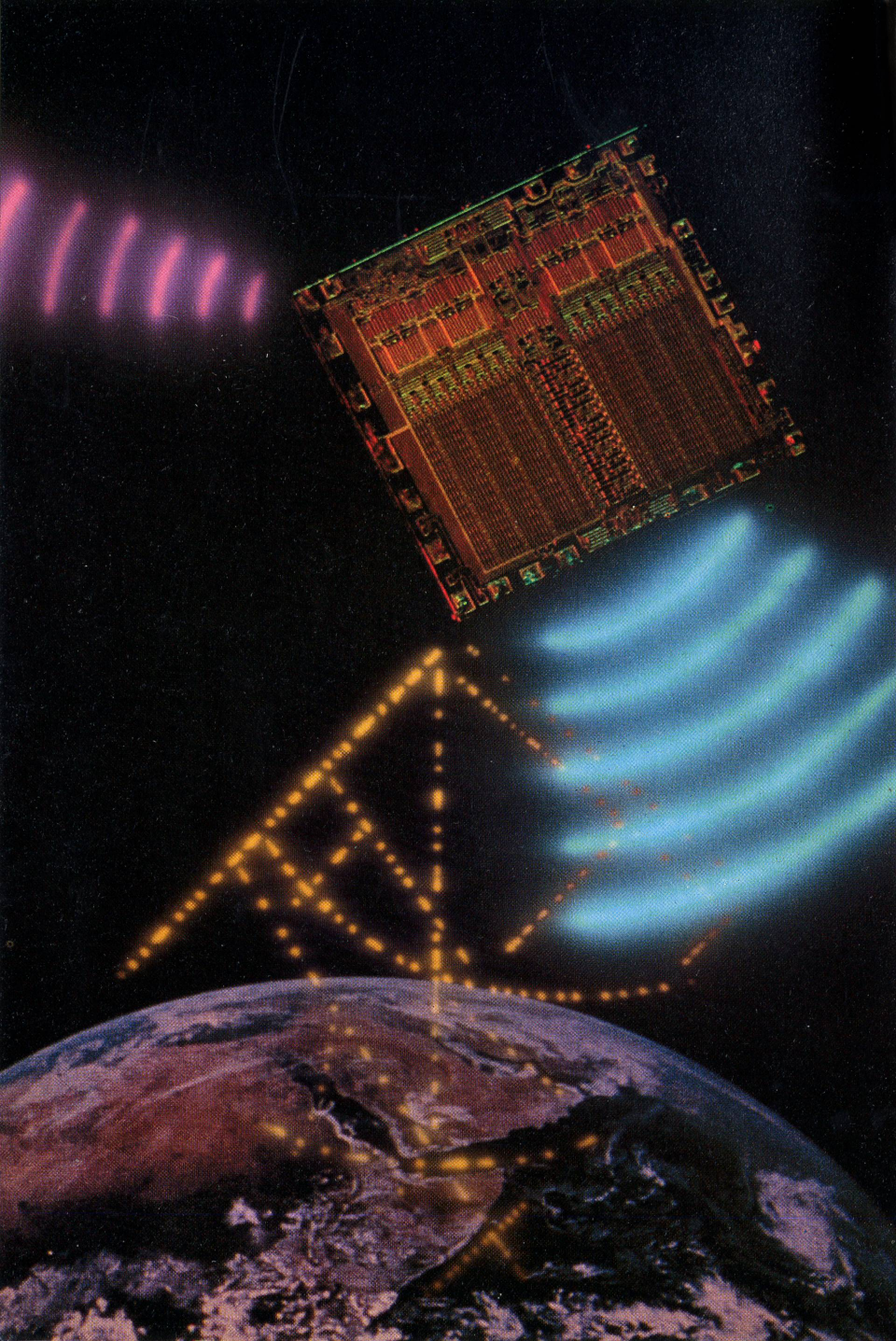
117 Dentro e fuori

143 Dal codice binario al linguaggio naturale

165 Oggi e domani

175 Glossario

188 Bibliografia

189 Indice analitico

A modo di introduzione

Un personal computer è, innanzitutto, un computer. Ciò che lo differenzia dai computer senza ulteriori specificazioni è quell'aggettivo, "personale": il quale, peraltro, sta solo a indicare come si tratti di apparecchiature adatte all'uso da parte di una persona sola, per dimensioni e soprattutto per prezzo, a differenza delle grandi apparecchiature che per dimensioni e prezzo possono essere possedute e utilizzate da società, organizzazioni o istituzioni. La differenza è sostanzialmente pratica, e non teorica: alla base, il personal computer che possiamo permetterci di acquistare e tenere in casa si fonda sugli stessi principi e funziona in modo analogo ai supercomputer che alla NASA vengono usati per il controllo delle missioni spaziali. Anche le differenze di dimensioni e prezzo, peraltro, sono vaghe e del tutto contingenti: in fondo, il personal che possiamo avere in casa oggi è molto piú piccolo e molto meno costoso di un grande computer di una quindicina di anni fa, ma ha una potenza che è dello stesso ordine di grandezza! E siccome l'evoluzione tecnologica in questo campo continua a ritmi impressionanti, il confine fra i computer personali e quelli che non lo sono continua a spostarsi verso l'alto.

Dire, dunque, che un personal computer è un computer non è solo una banalità nella miglior tradizione di monsieur de Lapalisse, ma nasconde un'idea di fondo che forse non è affatto evidente a prima vista: non c'è frattura fra il mondo dell'informatica personale e quello dell'informatica *tout court*. Le tecniche sono sempre le stesse, i concetti sono i medesimi, imparare a conoscere un personal computer significa imparare a conoscere che cos'è un computer. E proprio da qui vogliamo partire.

La Terra è sempre piú piccola: informatica e microelettronica contribuiscono ad accrescere l'integrazione del pianeta.

Un computer è una macchina

Prendete la macchinetta del caffè. Da sola non fa proprio nulla. Dovete mettere acqua nel contenitore inferiore, polvere di caffè nel filtro e chiuderla bene. Poi dovete metterla sul fornello e accendere il fuoco. A quel punto si mette in moto una serie di trasformazioni: l'energia chimica contenuta nel gas che la rete pubblica porta fino al fornello si trasforma in energia termica che viene comunicata alla macchinetta, provocando l'innalzamento della temperatura dell'acqua fino all'ebollizione. Un'altra serie complessa di trasformazioni porta l'acqua attraverso il filtro e infine nella camera superiore della macchinetta, ormai sotto forma di caffè: siamo cosí abituati a fare e a veder fare caffè che non ci soffermiamo piú a pensare alla complessità delle operazioni e delle attività fisiche e chimiche che stanno alla base di questa bevanda. Ma quel che è importante ricordare è che una macchina funziona solo se le viene fornita energia: una macchina trasforma energia in lavoro.

Anche un computer è un macchina: molto piú raffinata e complessa della

macchinetta per il caffè, ma sempre una macchina. E ha bisogno di energia, per funzionare: energia elettrica, per l'esattezza, che può essere l'elettricità di rete nel caso della maggior parte delle apparecchiature, oppure quella fornita da una serie di pile, nel caso dei computer portatili.

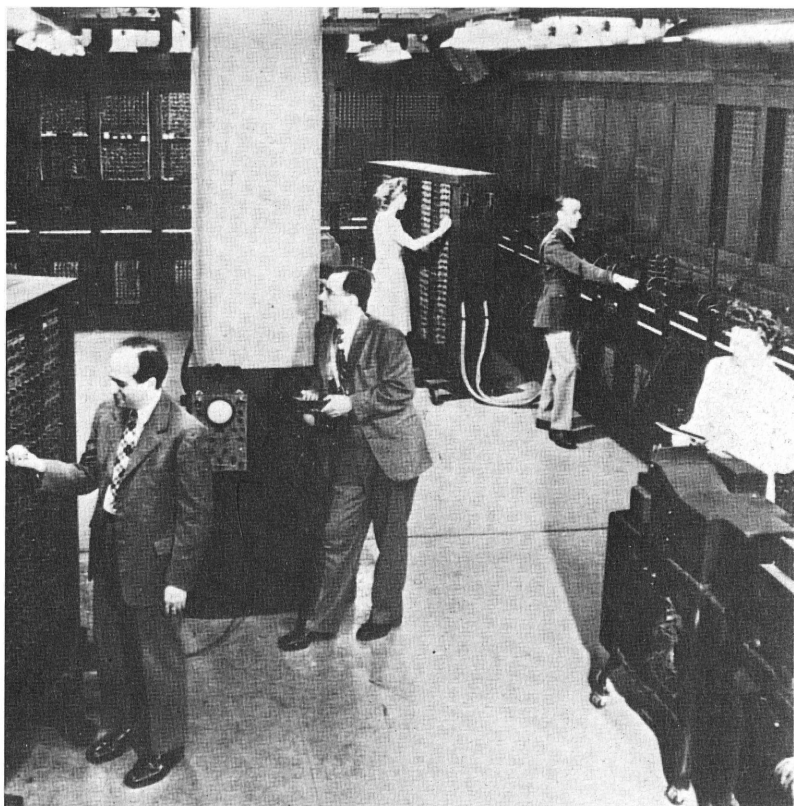
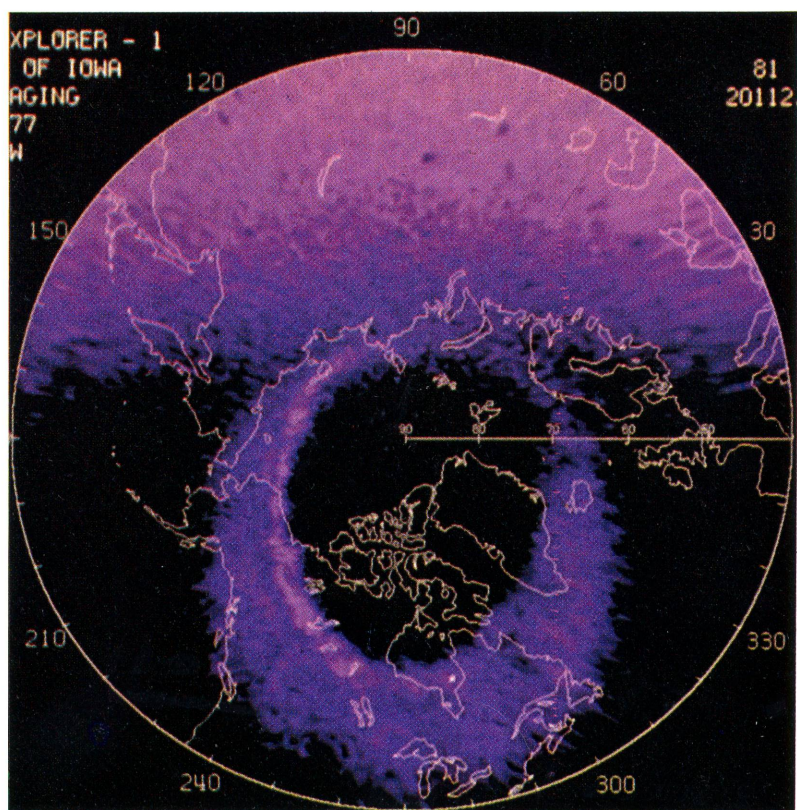
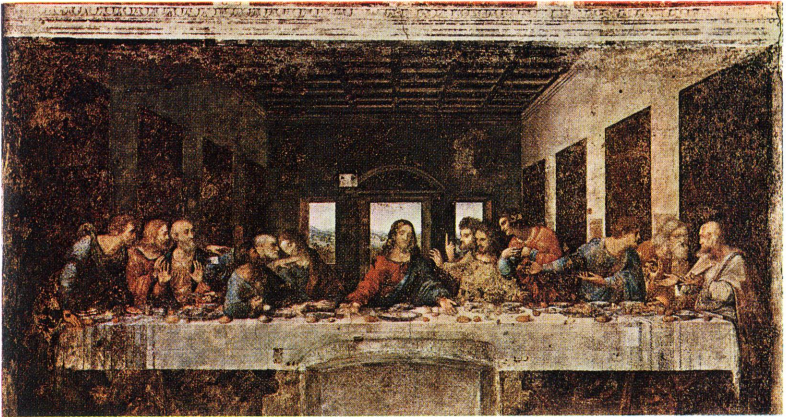
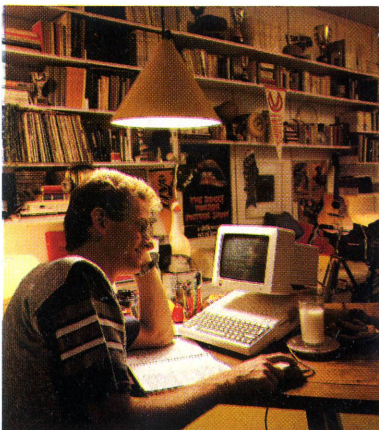
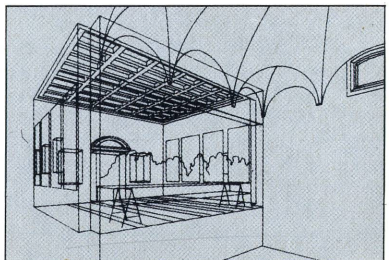
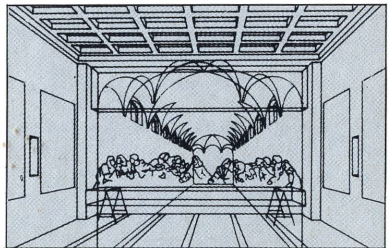
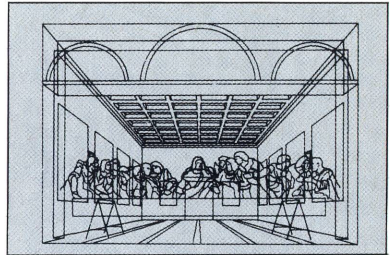
Il computer elabora informazioni

Sugli spalti del castello si è accesa una lanterna: è il segnale. Le porte sono aperte: gli assedianti potranno entrare e prendere alla sprovvista la guarnigione, conquistando la piazza. Potrebbe essere uno spunto per un gioco di avventure, ma immaginatelo come un episodio storico. Che cosa è successo sugli spalti? Qualcuno dall'interno fa chiaramente il gioco degli assedianti: ha predisposto che a una cert'ora le porte del castello siano aperte per rendere possibile l'ingresso indisturbato, e invia un segnale convenuto. "Se a mezzanotte vedrete accendersi una lanterna sugli spalti, la via è libera".
Anche in questo caso è in gioco energia: accendere la lampada significa trasformare l'energia chimica dell'olio combustibile in energia termica e luminosa. L'effetto però è del tutto sproporzionato alla piccola quantità di energia in gioco: uomini che si muovono, un castello conquistato, magari messo a ferro e fuoco, magari una guerra vinta, i precari equilibri politici di una regione medievale completamente sconvolti... altro che la macchinetta del caffè!
Nella sua semplicità, anche una lanterna è una macchina, una macchina per fare luce, ma in questo caso è usata come un mezzo di comunicazione di informazioni. Anche nell'informazione è sempre in gioco energia, ma il rapporto fra energia e informazione è molto diverso da quello che intercorre tra energia e lavoro in una macchina convenzionale. L'informazione non è importante per il suo supporto fisico, ma per ciò "per cui sta". La fiammella della lanterna è piccola cosa, ma l'informazione che comunica può essere sconvolgente. La fiammella, il supporto fisico, non è essenziale: la stessa informazione poteva essere comunicata con un messaggio scritto, con un urlo, con lo sventolio di una bandiera, con un gesto. Quel che importa è la convenzione, l'accordo sul significato: lanterna accesa significa via libera, lanterna spenta significa che la via non è libera, bisogna aspettare, non far nulla.
Le quantità di energia in gioco in un computer sono molto piccole: il consumo di elettricità di questa macchina non inciderà molto sulla bolletta, a fine mese! L'elettricità fornisce solo il supporto all'informazione, secondo schemi convenzionali, un codice convenuto. L'attività di un computer consiste nel manipolare segnali elettrici che rappresentano informazioni: la presenza di tensione in un certo punto o la sua assenza possono avere lo stesso significato dell'accensione o meno della lanterna.
La scelta della lanterna non è poi tanto peregrina: ci fornisce il tipo piú semplice di comunicazione possibile. Con la lanterna accesa o spenta possiamo comunicare solo un sí o un no: un'alternativa elementare. Quel sí o quel no rappresentano una quantità minima di informazione: un "bit" di informazione.
Si potrebbero usare due lanterne, per comunicazioni piú complesse. Una lanterna sulla torre di sinistra, una sulla torre di destra del castello. Quella di sinistra dice se la via è libera o meno, quella di destra se la porta aperta è quella sul davanti del castello o sul retro (potete inventare da voi qualche situazione piú avvincente, se questa vi sembra un po' fiacca). In ogni caso ciascuna lanterna comunica solo un sí o un no: ma aumentando opportunamente il numero della lanterne, si può arrivare a comunicare qualunque informazione, di qualunque complessità (in base a un codice opportuno).
Potete pensare a una serie di lampadine, ciascuna accesa o spenta da un interruttore. Qualcosa di simile succede sui centralini telefonici: c'è una piccola lampadina per ogni telefono collegato, che si accende se quel telefono è occupato. Supponiamo che i telefoni collegati al centralino siano otto. Se avessi-

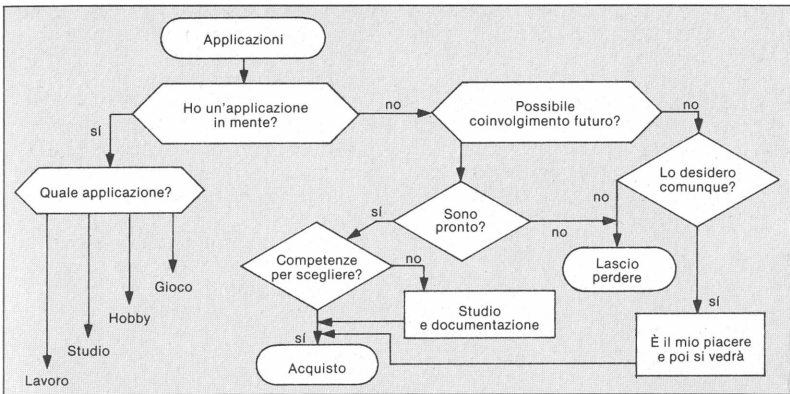
mo una lampadina sola, tutto quello che potremmo farle comunicare è se almeno un telefono è occupato; ma non riusciremo a comunicare quale (o quali) degli otto. Con due lampadine potremmo suddividere i telefoni in due gruppi: ogni lampadina ci direbbe se è occupato almeno un telefono di uno dei due gruppi. Ci vogliono proprio otto lampadine, una per ciascun telefono: allora in ogni momento possiamo sapere se c'è qualche telefono occupato e di quale telefono si tratta effettivamente. Se aumenta il numero dei telefoni è sufficiente aumentare il numero delle lampadine, e avremo sempre la situazione sotto controllo.

Un computer, in fondo, non è che un enorme numero di interruttori collegati insieme, ognuno dei quali controlla un "bit", una quantità elementare di informazione, un sí/no (passa corrente/non passa corrente). Tutto il suo funzionamento si basa su questo meccanismo binario: il meccanismo piú semplice, ma anche – si è scoperto – il piú economico ed efficiente per rappresentare l'informazione.

Si può benissimo usare un computer senza sapere affatto come funzioni internamente: molti lo fanno, esattamen-

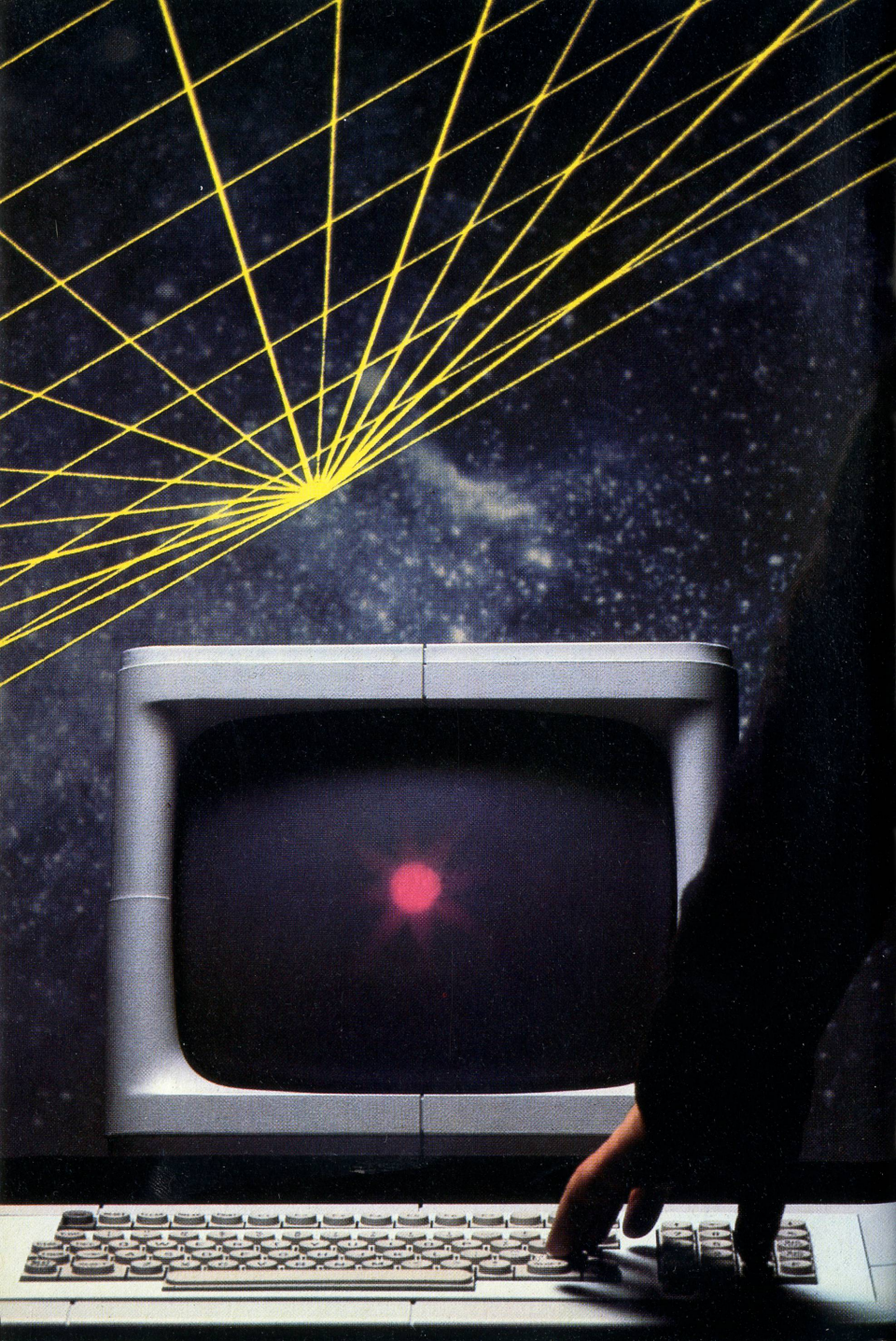
L'ENIAC, il primo computer costruito fra il 1943 e il 1945 occupava 160 metri quadrati, ma un "personal" da scrivania di oggi ha la stessa potenza!

I principi di funzionamento di un personal computer sono gli stessi dei grandi calcolatori, come quelli che aiutano gli studiosi di scienze della terra, per esempio a elaborare l'immagine a fronte, partendo da una foto aerea, per evidenziare l'estensione dell'aurora. Identità di principi non significa anche capacità di fare le stesse cose: questa è condizionata da altri fattori, come le dimensioni delle memorie di lavoro e di massa. Anche sui personal, però, sono accessibili tecniche grafiche come quelle utilizzate nel progetto di restauro dell'Ultima Cena di Leonardo: creazione ed elaborazione di modelli tridimensionali, con possibilità di presentazione automatica del soggetto da punti di vista differenti.

te come molti imparano perfettamente a guidare un'automobile senza sapere nulla dei particolari costruttivi del motore. Non c'è dubbio tuttavia che qualche conoscenza, almeno di carattere generale, sul funzionamento interno possa essere estremamente utile, se non altro per avere almeno qualche idea su quel che in effetti ci si può aspettare da un computer e quello che è del tutto impossibile (in teoria o allo stato attuale delle conoscenze).
Per inciso: il paragone con l'automobile è molto efficace e istruttivo. Lo useremo spesso, nel seguito: in fin dei conti, il personal computer promette di realizzare una rivoluzione, nel campo delle comunicazioni, non dissimile da quella che l'automobile personale ha innescato nel mondo dei trasporti (e di riflesso in tutta la società).

Il computer lavora sotto controllo di un programma

Elaborare informazioni non è tipico solo del computer: senza ricorrere al nostro cervello, elaboratore di informazioni per eccellenza, il mondo tecnologico offre molti esempi di macchine che elaborano informazioni. Senza andare troppo lontano, possiamo prendere le calcolatrici meccaniche o le piccole calcolatrici tascabili. Queste ultime, addirittura, sfruttano le stesse tecnologie elettroniche dei calcolatori e, in effetti, a prima vista potrebbe sembrare che le differenze tra calcolatrici e computer siano solo quantitative e non qualitative.
Ma una differenza essenziale c'è: una piccola calcolatrice tascabile è in grado di effettuare le quattro operazioni fondamentali e poco piú, ma sempre e solo quelle: non è possibile "insegnarle" a fare qualcosa di diverso sfruttando quelle sue capacità elementari. Un computer, al contrario, ha capacità elementari ancora piú semplici, ma è una macchina programmabile: di per sé non sa fare nulla, ma sfruttando le sue capacità elementari è possibile insegnarle a fare (quasi) qualsiasi cosa. Il programma è un po' come le istruzioni delle scatole di montaggio: se le seguite alla lettera, anche se non avete mai montato un mobile o un modellino d'aeroplano, con un minimo di pazienza potete riuscirci benissimo al primo tentativo. O, anche, è un po' come le ricette di cucina: vi dicono sostanzialmente tutto quello che dovete fare, e potete riuscire a cucinare un ottimo pollo alla cacciatora anche se non l'avete mai fatto prima. Questo perché tanto le istruzioni delle scatole di montaggio quanto le ricette di cucina fanno appello solamente a un bagaglio di conoscenze molto limitato, che si può presumere sia già patrimonio di chiunque si voglia cimentare in un esercizio di montaggio o di cucina, e guidano il lettore passo per passo, dettagliatamente, attraverso tutte le fasi.
Un programma per computer deve essere estremamente piú dettagliato di una ricetta o delle istruzioni di una scatola di montaggio: le conoscenze che costituiscono il bagaglio "culturale" e operativo della macchina sono davvero estremamente elementari, come trasformare uno 0 in un 1 o viceversa, spostare una successione di cifre binarie da una parte all'altra, sommare o sottrarre due numeri binari (ma in genere non rientra nelle capacità elementari di un computer moltiplicarli o dividerli: anche queste operazioni richiedono un programma!). Il principio comunque è il medesimo.
Il computer rimane dunque un macchina, e come tutte le macchine ha bisogno di energia e compie un lavoro; il lavoro che svolge è la manipolazione di segnali elettrici che rappresentano informazioni; e qualunque attività svolga è governata da un programma, una serie di istruzioni codificate.
Senza programmi un computer è solo un bel soprammobile: non fa proprio nulla. È come una automobile senza guidatore: non va da nessuna parte. Ma la sua caratteristica principale e la piú utile è proprio la versatilità che risulta da questo suo essere "tabula rasa": come una automobile non è predisposta per andare in un luogo ben preciso, ma può andare dovunque, cosí un computer può svolgere qualunque attività per la quale esista un programma che lo "istruisce". Ma ovviamente una automobile andrà

Il diagramma può aiutare a valutare il fattore chiave per il corretto acquisto di un computer: l'uso che se ne intende fare.

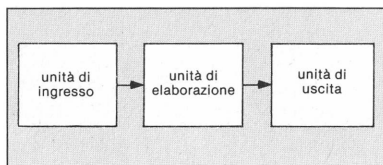
solo fin dove le strade, o almeno la natura del terreno, lo permettono: non scalerà una parete di sesto grado e si impantanerà nelle sabbie mobili. E anche per il computer esistono pareti di sesto grado e sabbie mobili: compiti troppo grandi per le sue capacità o addirittura impossibili teoricamente.

Imparare a guidare un personal computer

Per la maggior parte delle persone, una automobile è utile perché permette di spostarsi comodamente e rapidamente quasi ovunque: ma per questo è sufficiente avere una macchina e saperla guidare, non è necessario sapere come è fatta di dentro, né come ripararla in caso di guasto. Anche per il personal computer si può dire lo stesso: è utile perché permette di svolgere comodamente e rapidamente molte funzioni che, con carta e matita o altri mezzi tradizionali, costerebbero tempo e fatica; ma per questo è sufficiente saperlo usare (meglio ancora, saper usare i singoli programmi che servono), non è necessario sapere come funziona internamente o dove intervenire se si guasta.
In altre parole, come si può imparare a guidare un'automobile senza sapere nulla o quasi sul funzionamento del motore, è possibile imparare a "guidare" un personal computer senza conoscenze approfondite sulle sue strutture fisiche e sui meccanismi interni di funzionamento. Ecco il nostro punto di partenza.
Non c'è piú nessuno, però, oggi, che ignori che le automobili hanno quattro ruote, un motore, un volante, e che per andare hanno bisogno di benzina o di qualche altro combustibile: qualche conoscenza, almeno generica, sull'hardware è indispensabile ed è bene entri a far parte del patrimonio comune di tutti.
Per chi vuole passare a uno stadio successivo – non solo usare i programmi scritti dagli altri, ma intervenire con modifiche sui programmi degli altri, o addirittura scriverne di propri – le conoscenze debbono essere per forza piú estese. Spesso non è necessario ancora sapere perfettamente il funzionamento interno della macchina, tuttavia è indispensabile conoscere almeno un linguaggio di programmazione (i personal computer in genere sono corredati all'origine di qualche versione del BASIC, ma non è detto che questo sia necessariamente il linguaggio piú adatto per tutti e per tutte le applicazioni, anzi). Quanto piú raffinati si fanno i progetti di programmazione, tanto piú precisa deve essere la conoscenza della macchina: i compiti piú impegnativi richiedono quasi sempre anche un controllo piú dettagliato delle risorse fisiche.

Guidare il personal computer

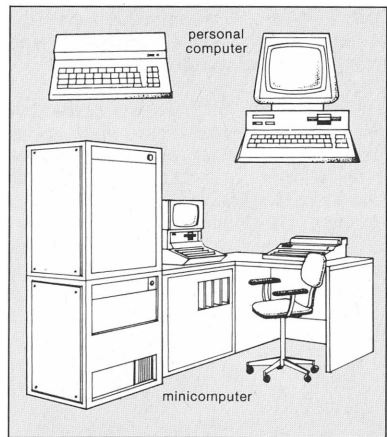
Un computer è dunque una macchina che elabora informazioni sotto controllo di un programma e, come tale, possiede una struttura generale molto semplice, sul piano concettuale. Qualunque computer ha almeno una unità di input, attraverso la quale possono essere inseriti programmi e dati; una unità centrale di elaborazione, nella quale si svolgono tutti gli effettivi processi di manipolazione delle informazioni; almeno una unità di output, che permette l'uscita dei risultati dell'elaborazione, in una forma comprensibile per l'operatore.
Se entriamo in un negozio di personal computer, ci rendiamo subito conto che non è difficile identificare in qualunque macchina queste unità funzionali essenziali: la normale unità di input per un personal computer è una tastiera, dall'aspetto molto simile alla tastiera di una macchina per scrivere, con tasti per lettere, numeri, simboli speciali e alcuni tasti (spesso diversificati per la forma, la disposizione o il colore) che svolgono operazioni "di controllo".
Le unità di output tipiche sono il video e la stampante: il video può essere un vero e proprio televisore, oppure un monitor, che dall'aspetto non si diversifica da un apparecchio televisivo, ma internamente non possiede i circuiti per la ricezione di programmi via etere e in genere permette una visualizzazione piú nitida e dettagliata. L'uscita sul video è temporanea: quando si toglie l'alimentazione, tutto ciò che è visualizzato si perde. Per ottenere copie permanenti, ci sono le stampanti: macchine che riportano i dati in uscita "nero su bianco" (in alcune stampanti raffinate e costose, anche in colore). Il procedimento di scrittura può essere analogo a quello di una macchina per scrivere elettrica, con elementi di scrittura a pallina o a margherita, oppure essere realizzato con testine ad aghi (che producono i tipici caratteri a matrici di punti), o con altri metodi ancora. L'unità di elaborazione, la parte cruciale della macchina, è in genere anche quella meno appariscente e caratterizzante: una manciata di circuiti stampati, di fili e di componenti elettronici vari, racchiusa in un contenitore metallico anonimo, della forma di un parallelepipedo per lo piú.
Per avere un computer funzionante, questi elementi sono indispensabili: almeno un'unità di input, un'unità di elaborazione, un'unità di output. Come siano fatte, come siano combinate è poco importante. Si trovano modelli di personal computer che incorporano in un unico contenitore tastiera e unità centrale (è il caso piú tipico per i piccoli elaboratori da casa): altri con tastiera separata, ma video incorporato nello stesso contenitore dell'unità di elaborazione; computer compatti con tutti gli elementi in un'unica unità fisica; altri con tutti gli elementi separati. Queste sono scelte puramente commerciali, che non hanno a che fare con la struttura concettuale di un computer. Per produrre un oggetto meno costoso, nel caso di un piccolo home computer, il produttore

Si può usare un personal computer come si usa un'automobile: si può imparare a guidarla senza sapere come è fatta dentro.

Visto sotto il profilo piú generale possibile, un computer è una macchina che elabora informazioni sotto controllo di un programma, e la sua struttura astratta è costituita da tre unità, rispettivamente per l'ingresso, l'elaborazione e l'uscita delle informazioni. Una tastiera è la piú comune fra le unità di ingresso; video e stampanti sono tipiche unità di uscita; le unità a dischi, per la memorizzazione sul lungo periodo, svolgono il doppio ruolo di unità di ingresso e uscita.

sceglie di integrare tastiera e unità centrale in un unico contenitore il piú semplice possibile e lascia la possibilità di utilizzare il televisore domestico come unità di output; nel caso di una macchina professionale, il costruttore si preoccupa magari piuttosto degli aspetti ergonomici, della comodità sul lavoro, dell'estetica generale di un ufficio: fornisce allora una tastiera separata, piú mobile e comoda, un'unità di elaborazione separata che può essere disposta a fianco della scrivania senza ingombrare il piano di lavoro, un monitor specifico (non si può pensare che in un ufficio esistano televisori adattabili), esteticamente in linea con gli altri componenti, orientabile per facilitare il compito del professionista o dell'impiegato...
È bene ricordare che tutti questi elementi sono importanti per una scelta oculata di un computer che risponda a pieno alle esigenze dell'acquirente, e non vanno sottovalutati: tuttavia non sono affatto essenziali al funzionamento di base della macchina.

La memoria di massa

Attraverso la tastiera si può comunicare con l'unità di elaborazione e attraverso il video o la stampante l'unità di elaborazione può a sua volta comunicare con l'utente, ma tutto ciò che avviene all'interno di questa semplice catena di comunicazione è transitorio e dipendente dall'alimentazione elettrica al sistema: ci serve ancora qualche dispositivo che permetta di conservare in forma permanente dati, programmi, risultati, in modo da poterli riutilizzare in futuro senza doverli ricostruire ogni volta da zero. È la differenza che sussiste tra il teatro e il cinema: a teatro ogni sera ha luogo una nuova recita, mentre gli attori cinematografici recitano una volta sola (o solo finché non producono un risultato soddisfacente) e la pellicola rimane poi indefinitamente, sempre uguale.
Il fascino della recitazione teatrale "dal vivo" non ha un corrispettivo nell'elaborazione elettronica: come tutte le analogie, anche questa ha i suoi grossi limiti. Se abbiamo preparato un programma per il nostro computer, dobbiamo inserirlo mediante la tastiera: una volta che è stato battuto tutto, lo si può "mandare in esecuzione" con un particolare insieme di dati, per ottenerne certi risultati, ma se poi si spegne la macchina il programma va perso irrimediabilmente, e la volta successiva lo si deve ribattere da zero. Siccome i programmi piú utili quasi sempre sono anche lunghi, non c'è proprio nessun elemento corrispondente alla "recitazione dal vivo": al contrario, si tratta di un'operazione noiosa,

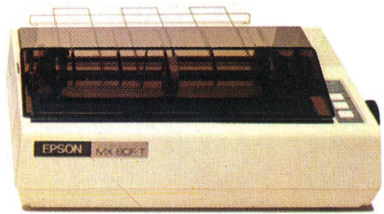
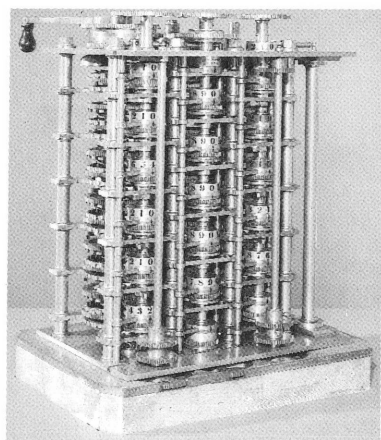
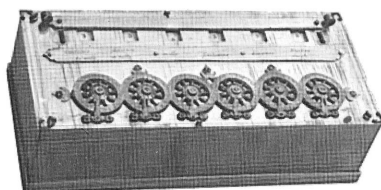
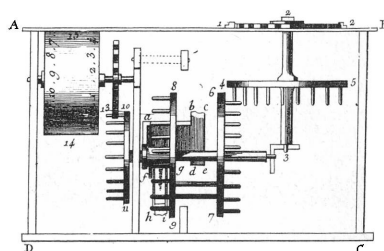
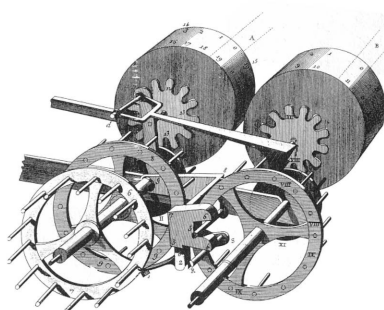
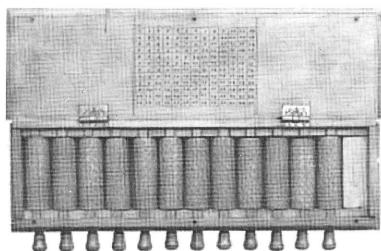
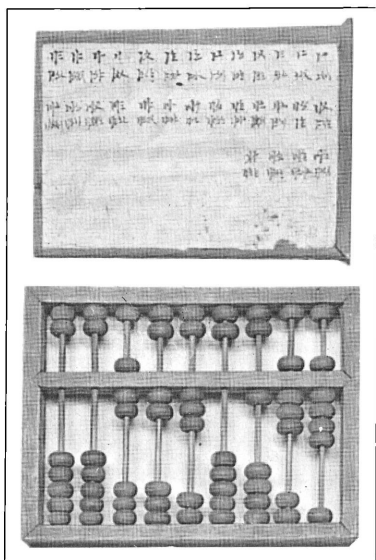
nella quale è facile commettere errori, e che richiede molto tempo.
È molto meglio disporre di qualche dispositivo che permetta di conservare programmi, dati, risultati, in una forma adatta al computer, che possa essere riutilizzata immediatamente, a distanza di tempo, senza dover ricominciare tutto il lavoro da capo. Quello che ci serve è un dispositivo che possa fungere, quindi, sia da unità di output, sia da unità di input: al quale si possano mandare informazioni dall'unità di elaborazione, ma dal quale poi esse possano essere recuperate successivamente e riportate nell'unità di elaborazione.
I dispositivi di questo genere sono le "memorie di massa" e sono analoghi a quelli che ci consentono di memorizzare e recuperare esecuzioni di brani musicali: nastri e dischi magnetici, con le relative unità di lettura e scrittura, registratori e drive. Una volta inserite in un computer, in fondo, le informazioni sono già sotto forma di tensioni elettriche: trasferirle a un supporto magnetico è un po' come registrare una trasmissione direttamente dalla radio (mediante un cavetto di collegamento, senza microfono), per poi riascoltarla a piacere.
I registratori che vengono usati come memoria di massa per i personal computer sono del tutto simili, all'aspetto, ai comuni registratori audio: in effetti, è possibile utilizzare un registratore audio che già si possiede, anche se i registratori studiati appositamente per i computer risultano molto piú efficienti (e piú affidabili, specie se digitali). Normalmente si usano registratori a cassette: quelli a nastro in bobina, molto costosi, sono usati con i grandi calcolatori.
I drive per dischi svolgono una funzione analoga a quella dei giradischi audio, ma, in piú, possiedono una testina di scrittura: è possibile "scrivere" informazioni sui dischi, oltre che leggerle. All'aspetto non assomigliano affatto ai giradischi: tutto quello che si vede non è altro che un contenitore a forma di parallelepipedo (eventualmente integrato nel contenitore dell'unità di elaborazione) con una fessura o uno sportellino in cui viene inserito il disco, di dimensioni piú

piccole dei 33 giri audio e di materiale piú flessibile. Esistono anche altre forme di memoria di massa, ma per ora ci basta: questi sono gli oggetti tipici che si possono vedere comunemente entrando in un negozio specializzato. Torneremo sull'argomento, piú dettagliatamente, nel capitolo terzo.

Modo immediato e modo differito

Prendiamo un personal computer molto semplice, costituito da una tastiera, un'unità di elaborazione, un video. Che cosa può fare? Dipende. Non tutte le macchine sono uguali. Un computer elabora informazioni sotto controllo di un programma: abbiamo visto di quali unità debba essere composto per poter svolgere questa funzione: ci siamo preoccupati di avere un'unità di input per dati e programmi, un'unità di output per vedere i risultati e un'unità di elaborazione. Ma manca una cosa: il programma o i programmi. Il computer non fa assolutamente nulla senza un programma di qualche tipo; in effetti, non è in grado di riconoscere neppure quale tasto abbiamo premuto sulla tastiera se non dispone di un programma che gli "dica" come farlo. Di piú: non è in grado nemmeno di "aspettarsi" che dalla tastiera gli arrivi qualche cosa. È difficile, forse, abituarsi a quest'idea, ma un

computer è proprio una tabula rasa: è estremamente ricettivo, ma bisogna insegnargli proprio tutto da zero.
Non spaventatevi: le ditte costruttrici di calcolatori si preoccupano di semplificarci la vita. Esiste una serie di funzioni elementari che comunque il computer sarà chiamato a svolgere: per esempio, tipicamente, quella di accettare segnali dalla tastiera o di inviare a sua volta segnali al video, di predisporsi in questo modo di funzionamento non appena viene acceso, e via dicendo. Queste funzioni sono preprogrammate dal costruttore e memorizzate una volta per tutte, in forma permanente, nella macchina, in particolari circuiti elettronici di memoria (le memorie ROM). Cosí, appena accendiamo il computer, i programmi conservati in queste memorie ROM vengono "caricati" nell'unità di elaborazione e ne controllano il comportamento, dicendo alla macchina quali operazioni deve fare.
I programmi conservati in forma permanente all'interno della macchina, nelle memorie ROM, sono diversi a seconda del computer e della casa costruttrice: è un fattore di cui si deve sempre tener conto, dal momento che influisce ovviamente anche sul prezzo del prodotto. Il caso piú semplice, per l'utente, è che il costruttore abbia dotato la macchina non solo di alcune funzioni di base, come un programma che si limiti a predisporre tutto per accettare input elementari dalla tastiera e visualizzare su video la comunicazione con l'utente, ma anche di programmi piú raffinati, come un programma che governi tutti i componenti del sistema e un programma "traduttore" che sia in grado di interpretare comandi, istruzioni, informazioni comunicate attraverso la tastiera in un linguaggio di programmazione abbastanza facile da usare: il BASIC, nella stragrande maggioranza dei casi.
Se il personal computer che abbiamo

Due configurazioni per un personal computer: in questa pagina una versione minima, adatta per l'uso domestico, con unità centrale e tastiera integrate, televisore come unità video e registratore a cassette per la memoria di massa (il computer è un Commodore VIC 20). Nella pagina a fronte, una configurazione dalle caratteristiche piú evolute, ancora con unità centrale e tastiera integrate, ma monitor (anziché televisore) e stampante come unità di uscita, e due drive per dischetti flessibili come unità di memoria di massa (si tratta di un modello Apple II, con stampante Epson).

davanti ha memorizzato su ROM il programma interprete del BASIC, questo verrà caricato automaticamente dal programma piú semplice (il programma di "lancio" o di "bootstrap") di cui abbiamo parlato prima, al momento dell'accensione, e saremo subito in grado di sfruttare utilmente la macchina.
Supponiamo di essere fortunati (non è difficile, in realtà) e che il personal computer che abbiamo di fronte sia stato fornito dal costruttore di questa dotazione abbastanza ricca di programmi in ROM. Basterà allora accendere il computer e subito saremo in grado di utilizzarlo. Premiamo l'interruttore di accensione e nel giro di pochi secondi lo schermo ci presenta la macchina: in genere compare una scritta che annuncia la disponibilità del BASIC (e precisa di quale fra le tante versioni del linguaggio si tratti), al di sotto della quale è visualizzato un simbolo speciale, il "prompt". Il "prompt", o simbolo di "pronto", indica che la macchina è predisposta e in attesa di istruzioni da noi. Accanto un segno lampeggiante, un rettangolino o una lineetta: è il cursore, che indica la posizione sullo schermo in cui verrà visualizzato il risultato della prossima operazione. Se premiamo un tasto qualunque sulla tastiera, vedremo comparire sul video, dove ora si trova il cursore, il carattere corrispondente. In questo caso il video funge da "eco" o da "retroazione", visualizzando l'input e permettendoci in tal modo di verificarne la correttezza.
In queste condizioni, possiamo usare il nostro computer piú o meno come una calcolatrice elettronica, con qualche piccola differenza. Proviamo a chiedergli di fare per noi qualche somma, per esempio di sommare 123 e 254. Facciamo come su una calcolatrice: scriviamo 123 + 254. Per concludere l'immissione di qualunque dato o comando, bisogna poi premere il tasto RETURN, che occupa sulla tastiera la posizione usualmente riservata, nelle macchine per scrivere, al tasto di andata a capo. Qui svolge quella funzione e, in piú, segnala all'unità di elaborazione che si è conclusa una fase di inserimento. Fino a quel momento, la macchina si limita a "prendere atto" dell'inserimento di alcuni caratteri; la pressione del tasto RETURN ha il significato di "ho finito: quello che precede è un'istruzione completa". Il programma interprete del BASIC svolge la sua funzione di traduttore, trasforma quanto abbiamo battuto in istruzioni comprensibili alla macchina, che poi provvede a eseguirle. (Non su tutti i computer il tasto che indica la conclusione di un'istruzione è etichettato RETURN: può essere chiamato anche ENTER, oppure può portare semplicemente un simbolo, per esempio la freccia dell'a capo come sulle macchine per scrivere: sono tutte differenze inessenziali, dovute solo alla totale mancanza

Il calcolatore elettronico è l'ultima tappa nella storia degli ausili per il calcolo, che parte dall'antichità con l'abaco (in alto un dispositivo di calcolo del 3000 a.C., subito sotto un esemplare moderno di abaco, del XIX secolo), ma ha avuto un impulso particolare nell'era moderna, con la rivoluzione scientifica prima, poi con quella industriale (e l'acquisizione di un ruolo centrale, da parte di scienza e tecnica, nei processi produttivi). Qui sopra, la macchina di Schott (1668), basata sul principio dei bastoncini di Nepero; a destra in alto, la "pascalina" di Blaise Pascal (1652). Qui a destra, la macchina alle differenze di Charles Babbage, in cui sono già operanti i principi di base dei calcolatori moderni.

Come già aveva intuito Ada Lovelace (figlia di Lord Byron e assistente di Babbage), il calcolatore non è solo una macchina per calcoli aritmetici, ma piú in generale per l'elaborazione simbolica. Una fra le prime applicazioni in tal senso fu l'elaborazione dei risultati del censimento degli Stati Uniti del 1890 con la macchina tabulatrice di Herman Hollerith (qui sotto; a destra un estratto del brevetto).
Tutto il potenziale di queste idee è stato sfruttato solo negli ultimi decenni, grazie alla tecnologia microelettronica e alle possibilità di integrazione su larga scala dei circuiti (in basso è schematizzato il processo di fabbricazione dei chip).

progetto

definizione delle specifiche funzionali → progettazione del circuito

fabbricazione dei microcircuiti

generazione delle maschere → prima maschera

sequenza di processi fotolitografici e chimico-fisici

ultima maschera

fetta di silicio ossidato → delineazione ossido → delineazione piazzole di connessione → collaudo dei microcircuiti

incapsulamento e collaudo

separazione piastrine → circuito singolo → saldatura piastrina e fili di connessione → incapsulamento → collaudo finale

di standardizzazione in questo campo.)
Bene: abbiamo battuto 123 + 254 e premuto il tasto RETURN. Immediatamente sulla riga inferiore, sul video, ricompare il segno di pronto, segnalandoci che il computer è disponibile ad accettare dati o comandi. Ma non ci ha risposto! Noi volevamo sapere il risultato!
Già, ma non glielo avevamo detto! Non bisogna mai dimenticare che il computer non pensa per noi e che bisogna sempre dirgli tutto quello che deve fare. Se gli diciamo solo di sommare 123 e 254, lo fa e basta: se vogliamo anche vedere il risultato, dobbiamo chiederglielo esplicitamente. Nel caso del BASIC (e di gran parte degli altri linguaggi di programmazione), il comando che abbiamo dato deve essere corretto in questo modo:

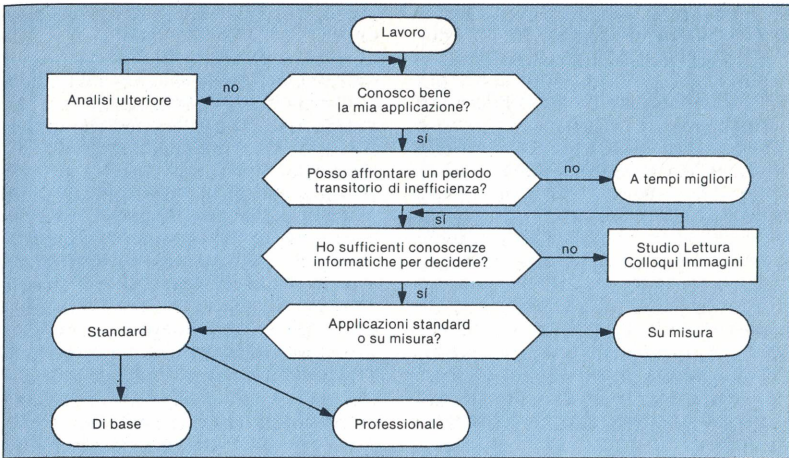
```
PRINT 123 + 254
```

dove la parola PRINT è una parola inglese adottata dal BASIC e significa "stampa" o, piú genericamente in questo contesto, "visualizza" (su monitor, non necessariamente su una stampante). Purtroppo i linguaggi con cui si può comunicare con il computer sono stati tutti formulati negli Stati Uniti e modellati sulla lingua inglese, che rimane un po' la lingua "ufficiale" dell'informatica: nel BASIC, in particolare, tutti i comandi sono o parole inglesi o abbreviazioni di parole inglesi. Fortunatamente i linguaggi con cui si comunica con il computer non sono molto ricchi e anche per chi non conosce l'inglese non c'è molto da imparare.
Il nostro nuovo comando, dunque, dice al computer qualcosa come "visualizzami il risultato della somma di 123 e 254". Proviamo a batterlo e a premere, poi, RETURN. Questa volta qualcosa è successo: in un baleno, sulla riga successiva, è comparsa la scritta 377, poi è stato visualizzato nuovamente il "pronto" con il cursore lampeggiante.
Con questo stesso criterio possiamo ottenere i risultati di operazioni un po' piú complesse, operando sostanzialmente come su una calcolatrice tascabile. Attenzione però ai simboli: il segno di moltiplicazione, sulla tastiera di un computer, è sostituito dall'asterisco, per la divisione si deve usare la barra (/) e per l'elevazione a potenza si usa il segno ^ (o la freccia con la punta verso l'alto, su alcune macchine). Sono piccole varianti a cui si deve fare l'abitudine, ma non comportano nessuna scomodità: sono dovute al fatto che il simbolo usato normalmente per la moltiplicazione qui potrebbe facilmente confondersi con la lettera "x" e che la rappresentazione è sempre lineare, mentre normalmente l'esponente nell'elevazione a potenza viene scritto, manualmente, come un indice sovrascritto.
Se fosse tutto qui, ovviamente, non varrebbe la pena di acquistare un personal computer: esistono anche altre funzioni matematiche, come quelle trigonometriche, ma anche una buona calcolatrice tascabile scientifica è in grado di svolgere benissimo gli stessi compiti. Anche la precisione non è molto maggiore: usato in questo modo il computer fornisce di norma una precisione di poche cifre, una decina circa (valore variabile da macchina a macchina).
Si può fare un piccolo esperimento. Si può provare a scrivere:

```
PRINT "MI SEMBRA UNA BELLA MACCHINA"
```

e poi premere RETURN. Che cosa succede? Sulla riga successiva del video è comparsa la scritta

```
MI SEMBRA UNA BELLA MACCHINA
```

il che dimostra come il computer sia in grado di manipolare, non solo numeri, ma anche simboli linguistici. Non gli abbiamo chiesto di eseguire un'operazione molto complicata, solo di visualizzare una scritta elementare, ma è già una cosa che una calcolatrice tascabile non può fare.
Ancora una volta, però, se fosse tutto qui non sarebbe molto. Quello che abbiamo "assaggiato" è solo un modo di funzionamento del computer, il modo "immediato", in cui alla macchina vengono forniti comandi diretti, che vengo-

La scelta dei programmi per applicazioni professionali non è facile: il software è molto piú importante dell'hardware.

no eseguiti immediatamente. Il BASIC permette di lavorare in questo modo, comodo per effettuare operazioni semplici, ma anche – e soprattutto – nel modo differito, o modo programma. Il computer lavora sempre con qualche programma, l'abbiamo già detto e lo ripeteremo molte volte: anche quando usiamo il BASIC in modo immediato, in realtà il computer è controllato da un programma abbastanza complesso che traduce in una forma comprensibile all'unità centrale quanto noi scriviamo alla tastiera e provvede a far eseguire le operazioni che vogliamo. Per noi, tuttavia, questo livello non è visibile: sappiamo che c'è, ma non lo usiamo sotto questo aspetto. Usiamo comandi che ci appaiono come comandi immediati, nell'esempio precedente e quel che succede dentro la macchina ci è "trasparente". Però, a nostra volta, possiamo costruire dei programmi sfruttando le capacità del BASIC, per usare la macchina in modo differito. Con il BASIC, per scrivere programmi è necessario numerare le righe di istruzioni con un numero progressivamente crescente, che va posto all'inizio della riga.

Per esempio, possiamo riprendere la nostra frase. Se scriviamo:

100 PRINT "MI SEMBRA UNA BELLA MACCHINA"

e diamo il RETURN, non succede nulla. Ricompare semplicemente il segno di pronto sulla riga successiva. Poiché la prima cosa che abbiamo scritto sulla riga era un numero, l'interprete BASIC ha riconosciuto questa riga come una riga di programma e si è limitato a memorizzarla e a conservarla. Perché succeda qualcosa, ora, dobbiamo scrivere RUN e premere RETURN. Sulla riga successiva comparirà la visualizzazione della frase:

MI SEMBRA UNA BELLA MACCHINA

e, una riga sotto, ancora il segno di pronto. Il nostro brevissimo programma rimane in macchina, pronto per essere eseguito ogni volta che scriviamo RUN: provare per credere!
Per quanto semplice, questo è già un programma! Utilizzando tutte le risorse del BASIC se ne possono scrivere ovviamente di molto piú complessi e di molto piú utili: vedremo meglio nel capitolo 5 le caratteristiche di questo linguaggio di programmazione.

Per ora possiamo immaginare solamente la possibilità di redigere programmi piú complessi: ma già scrivendo un'unica riga di programma ci si può rendere conto di una cosa importante. Se non è facile costruire un buon programma, non è facile neanche semplicemente trascriverlo alla tastiera. Basta un nonnulla, un piccolo errore di battitura, e il programma può non funzionare piú, oppure funzionare in maniera diversa da quel che ci si aspetta. Un'altra persona può capirvi anche se pronunciate male una parola, o non vi esprimete molto correttamente, perché può sfruttare tutta la sua esperienza e la sua intelligenza e intuire le vostre intenzioni comunque; ma un calcolatore non lo può fare, vi prende sempre perfettamente alla lettera.
Per questo non è pensabile dover riscrivere ogni volta un programma quando lo si deve usare: è indispensabile memorizzarlo su un supporto autonomo e permanente, un dischetto o una cassetta. Ma, cosa ancor piú importante, non è nemmeno pensabile che ciascuno debba costruirsi da sé tutti i programmi di cui può aver bisogno: per compiti complessi, un programma può costare giorni, quando non mesi di lavoro assiduo. Un buon programma è come un buon libro: si deve pensare che cosa scrivere, preparare un piano di lavoro, scrivere, leggere, correggere, rileggere... non si rivela mai un'impresa di poco conto.
Esistono programmi già predisposti per la guida dei calcolatori personali: sono come i libri che si comprano in libreria. Non c'è bisogno di scriverli, basta leggerli. E usarli è facile, se sono fatti bene. Purtroppo, bisogna aggiungere, come ci sono libri belli e libri meno belli ci sono anche programmi piú o meno buoni e, come un libro può piacere o meno, anche un programma può risultare piú o meno confacente alle necessità dell'utente: per questo, prima di acquistare un programma già confezionato, soprattutto nei casi in cui il costo è elevato, è bene provarlo (un buon rivenditore è sempre attrezzato per consentire all'utente una prova dei programmi, con l'aiuto di un tecnico). In fondo, non si fa cosí anche con i vestiti?
E se non si trova quello della misura giusta, meglio procrastinare l'acquisto, piuttosto che comperarne uno di un'altra taglia. Usare un programma preconfezionato è comunque semplice: chi l'ha scritto si è preoccupato di dotarlo di tutti gli elementi che servono a sveltirne l'utilizzazione. Per partire in genere non c'è bisogno di quasi nulla: il procedimento però è un po' diverso a seconda del dispositivo di memoria di massa che si possiede (se è un registratore a cassette o un'unità a dischetti). Il caso decisamente piú elementare è quello dell'unità a dischetti e, per far scomparire quel poco di paura che può esservi rimasta all'idea di dover affrontare un personal computer, eccovi una "lezione di guida" con un programma di tipo commerciale su un tipico personal computer dotato di unità centrale, tastiera, video e un'unità di lettura e scrittura di dischetti. Il programma è uno dei piú diffusi in tutto il mondo per l'elaborazione di testi: trasforma cioè il computer in una raffinatissima "macchina per scrivere", ma con capacità che superano di gran lunga quelle di una comune macchina per scrivere. La scelta è caduta su questo programma perché ne esistono versioni per quasi tutte le macchine in commercio al momento in cui scriviamo, perché è in commercio già da anni (e questo ci assicura che non si tratti di un successo effimero) e, infine, perché l'elaborazione di testi (o word processing, nella terminologia anglosassone) è fra le funzioni piú comuni: quasi tutti hanno bisogno prima o poi nella vita di scrivere qualcosa e molti hanno avuto modo di usare già una macchina per scrivere (cosí sarà possibile un confronto, almeno su un punto, fra uno strumento tradizionale e il nuovo tipo di strumento).
A computer spento, inseriamo il dischetto con il programma nel drive e richiudiamo lo sportellino. È buona regola utilizzare una copia del disco, non l'originale: l'originale andrebbe duplicato immediatamente e riposto lontano da ogni possibile rischio (polvere, cenere

di sigaretta, gocce di caffè, sorgenti di caldo eccessivo e via dicendo). Supponiamo dunque di avere una copia già pronta (duplicare è un'operazione semplice, nel peggiore dei casi solo noiosa) e, per comodità, di memorizzare il testo sul medesimo disco (è meglio, di norma, archiviare su un dischetto di lavoro distinto).

Inserito il dischetto e chiuso lo sportellino del drive, si accende il computer: il programma verrà caricato automaticamente. (Per ottenere questo risultato, il dischetto con il programma è stato preparato opportunamente: non per tutte le macchine e non per tutti i programmi commerciali le cose stanno in questo modo: in molti casi, dopo accesa la macchina, è necessario digitare il nome del programma e premere RETURN. In genere le cose non sono piú complicate di cosí.) Il nostro programma ci chiede la data e l'ora, che inseriamo nel formato richiesto (gg-mm-aa, cioè numero del giorno, numero del mese e due ultime cifre dell'anno, in quest'ordine e separati da un trattino, per la data: hh:mm:ss, cioè ore, minuti e secondi separati da due punti per le ore).

Finalmente sono finite le formalità iniziali, e possiamo cominciare a lavorare: il programma ci presenta un "menú", un elenco di alternative fra le quali possiamo scegliere. A ogni alternativa corrisponde una lettera: per indicare l'alternativa prescelta è sufficiente premere il tasto corrispondente a quella lettera. Nella fattispecie, il programma accetta l'indicazione con lettere minuscole o maiuscole indifferentemente.

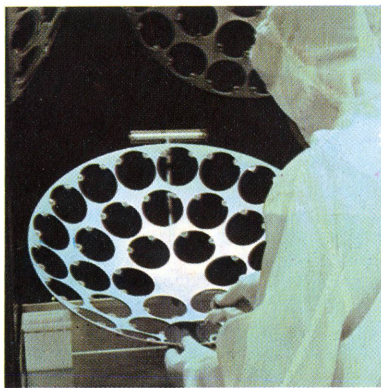
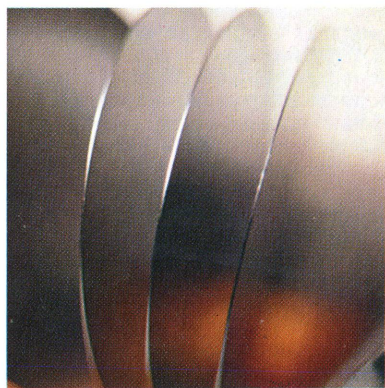
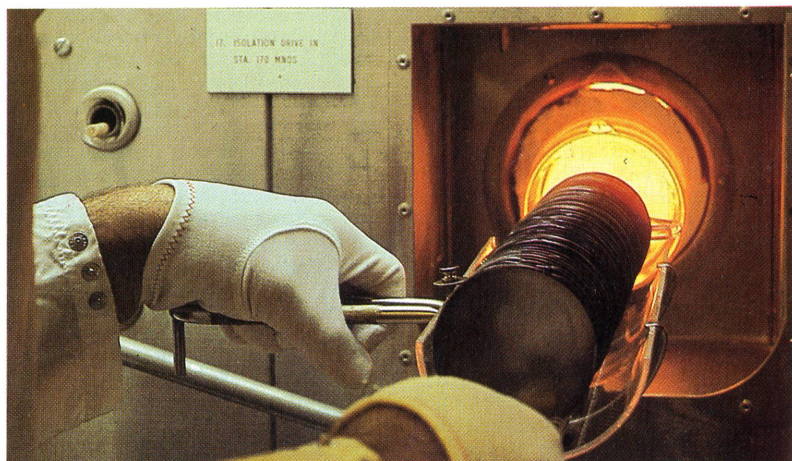
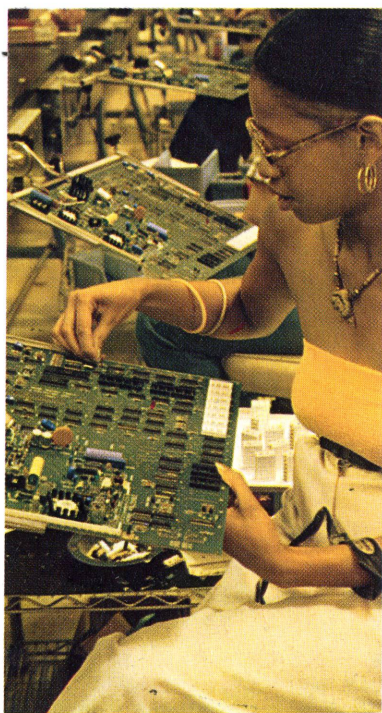
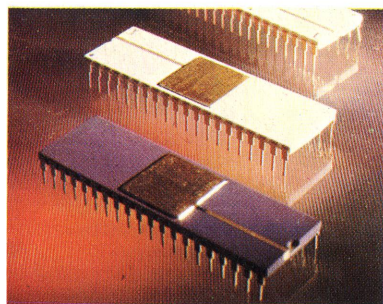
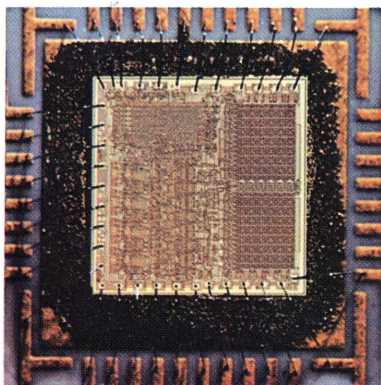
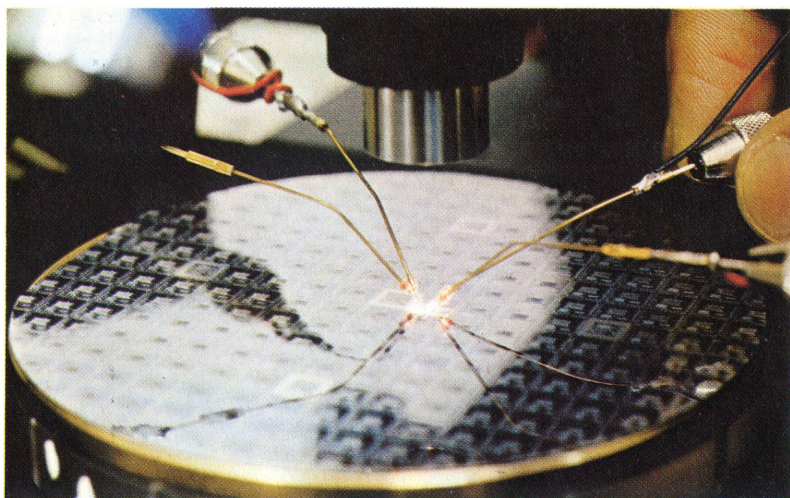
Il menú è fra gli strumenti piú efficaci per rendere agevole l'utilizzazione di un programma: non è necessario mandare a memoria un gran numero di comandi, è sufficiente leggere attentamente il menú e scegliere quel che si desidera. Il programma che stiamo usando permette poi all'utente esperto di fare a meno dei menú: una delle voci che si possono notare in questo menú "esterno" riguarda i livelli di aiuto e al livello di aiuto 0 il programma elimina la visualizzazione di tutti i menú. Il livello di aiuto massimo, il 3, è quello al quale il programma si situa automaticamente, se non gli viene chiesto esplicitamente altrimenti. Anche la tecnica dei valori standard per vari parametri, valori che il programma assume automaticamente se non gli viene detto esplicitamente il contrario, facilita di gran lunga il compito dell'utente, che non deve affatto preoccuparsi di molte cose. Solo quando diventa piú smaliziato, o quando le sue esigenze crescono, può andare alla scoperta delle altre possibilità offerte dal programma.

Il nostro obiettivo è creare un testo: il menú esterno ci suggerisce di premere il tasto D (per la creazione di un documento: l'opzione "non documento" va scelta quando si vogliono scrivere, per esempio, programmi per linguaggi compilati come il Pascal o il C). Appena premuto il tasto D, il programma ci chiede di dare un nome al nostro documento: con grande fantasia, possiamo chiamarlo PROVA. (Sono accettabili nomi con un massimo di otto caratteri, eventualmente seguiti da un punto o da un'estensione di altri tre caratteri.)

Incidentalmente, la parola *file* che compare sul menú è una parola inglese (si pronuncia "fàil") e significa "archivio". Si può immaginare un dischetto o qualsiasi supporto di memoria come un classificatore a cartelle sospese, di quelli che popolano tipicamente gli uffici: un file ha lo stesso ruolo di una cartella sospesa, ospita cioè una certa quantità di materiale, identificato da un nome (l'etichetta sulla cartella) e relativamente omogeneo (ma questo non è particolarmente importante). Un file dunque può essere costituito da una lettera, un programma, un libro o un capitolo di libro, un insieme di dati di varia natura, e via dicendo.

Il nome del file lo identifica univocamente: abbiamo dato il nome PROVA al nostro testo, e d'ora in poi questo nome sarà un po' la chiave per accedervi, qualunque sia l'operazione da eseguire sul file nel suo complesso.

Dato il nome al file-documento, il "menú esterno" scompare: ha svolto la sua funzione, ci ha portati dall'esterno all'interno del programma e lascia ora il po-

I componenti elettronici dei sistemi di elaborazione sono integrati su minuscole piastrine di silicio, i chip, la cui fabbricazione è un processo di alta tecnologia. La progettazione iniziale può avvenire al computer; da cilindri di monosilicio cristallino, maneggiati con cura per evitare impurità indesiderate, si ricavano fette circolari, montate su sostegni per le fasi successive e suddivise in quadratini. I circuiti vengono realizzati con tecniche analoghe a quelle di stampa; i collegamenti sono attuati con alluminio. I chip vengono montati su supporti ceramici, poi in contenitori protettivi di plastica; infine sono raggruppati su schede, per ottenere funzioni particolari.

sto a un secondo menú, il "menú principale". Sotto il menú una riga di trattini periodicamente interrotta da un punto esclamativo: ha la stessa funzione del tabulatore sulle macchine per scrivere, ci dice cioè dove sono posizionati il margine sinistro (l'estremità sinistra della riga), il margine destro (l'estremità destra della riga), i punti di arresto di tabulazione (i punti esclamativi). Automaticamente il programma ci dà righe di 65 caratteri, con margini fissati a 0 e 65 e punti di arresto di tabulazione ogni 5, fino al 55. Non bisogna preoccuparsi di nulla: è già tutto preselezionato. Abbiamo esigenze particolari, vogliamo margini diversi e arresti di tabulazione in posizione diversa? Nessun problema: basta premere contemporaneamente i tasti CONTROL e 0 e comparirà il "menú impostazione", con l'indicazione di tutti i modi in cui è possibile modificare l'impostazione standard.

Per noi ora l'impostazione standard va benissimo: vogliamo solo capire i meccanismi di fondo. Sotto la riga di tabulazione, lo schermo è vuoto: qui possiamo cominciare a scrivere quello che vogliamo. La lineetta lampeggiante è il cursore: indica il punto dove apparirà il prossimo carattere battuto. Inserire il testo ora è un'operazione non dissimile dalla battitura su una macchina per scrivere: si usa la tastiera nello stesso modo. Quando si arriva a fine riga, però, non c'è da preoccuparsi del margine. Proviamo a scrivere proprio questa frase:

Quando si arriva a fine riga, però, non c'è da preoccuparsi del margine.

Che cosa succede? Tutto normale fino all'ultima parola: i caratteri e gli spazi si dispongono nell'ordine in cui li inseriamo. Poi cominciano a scrivere "margine" e succede qualcosa di strano: quando battiamo la "r" il programma automaticamente ci porta a capo tutta la parola e allarga gli spazi della riga, in modo da chiudere esattamente i 65 caratteri prefissati. Non c'è bisogno di dare l'a capo a ogni riga come sulle macchine per scrivere: è il programma che si preoccupa di "contare" le battute e di sistemare le righe in modo da non superare mai i margini. Quando una parola non sta per intera sulla riga, il programma la porta automaticamente a capo e allarga gli spazi nella riga terminata per "giustificarla", come si dice con termine tecnico, cioè per allinearla a destra, oltre che a sinistra. L'a capo automatico viene spesso identificato con il nome di *word wrap*.

Il tasto di a capo (il RETURN) si usa solo quando si conclude un paragrafo e la riga rimane incompleta.

La modalità "giustificazione" è un'altra delle caratteristiche che il programma sceglie automaticamente, ma può anche essere eliminata: è possibile cioè far funzionare il programma "a bandiera". In questo caso il programma continua a mandare a capo la parola, quando non può stare intera in una riga senza superare i margini, ma non allarga piú gli spazi e lascia la riga cosí com'è. Il margine destro non sarà piú allineato, ma irregolare. Non c'è bisogno di optare una volta per tutte per una alternativa: il bello dei programmi di word processing è proprio la possibilità di modificare un parametro, alla fine del lavoro, e far modificare automaticamente di conseguenza tutto il testo.

Cosí è, per esempio, per i margini: si può scrivere con i margini 0-65 e alla fine decidere se è meglio avere righe piú lunghe: 0-75, per esempio. Nel caso del nostro programma, basta portarsi all'inizio del file e premere insieme i tasti CONTROL e B: il testo viene riorganizzato secondo i nuovi parametri. (CONTROL B ha valore solo per un paragrafo: a ogni paragrafo bisogna ripetere il comando. Altri programmi hanno comandi diversi, magari solo globali e non paragrafo per paragrafo come questo.)

Ci possiamo divertire a scrivere quello che vogliamo (vedi le illustrazioni delle pagine 30-35). Capita un errore di battitura? Nessun problema. Basta portare il cursore sul punto in cui si è commesso l'errore e correggere. Come si sposta il cursore? Con i quattro tasti con le frecce. La freccia in alto ce lo porta in su di una riga, la freccia in giú ce lo porta in basso di una riga, le frecce a sinistra e a

destra ce lo spostano di un carattere alla volta. (Guardando il menú si può vedere che gli stessi spostamenti possono essere realizzati con comandi opportuni, usando il tasto CONTROL insieme a un altro tasto. La macchina che abbiamo usato per il nostro esempio, dotata di tasti per lo spostamento del cursore, ci facilita il compito: il programma è abilitato a riconoscere la funzione di questi tasti.) Una volta posizionato il cursore, bisogna guardare in alto sullo schermo per vedere in che modalità siamo: se c'è scritto "INSERIMENTO", vuol dire che qualunque carattere battiamo andrà a inserirsi subito prima del cursore; se invece la scritta INSERIMENTO è assente, la modalità è quella di "sovrascrittura", cioè il carattere che batteremo andrà a sovrapporsi a quello su cui è posizionato il cursore, sostituendosi ad esso. La modalità inserimento è utile quando si è saltata una lettera o una parola intera; la modalità sovrascrittura è utile quando si è battuta una lettera sbagliata. Si può usare anche solo la modalità inserimento: basta, prima di battere la lettera o la parola giusta, cancellare la lettera o la parola sbagliata. Per cancellare, nel nostro caso, basta usare il tasto DELETE (*delete* in inglese significa proprio "cancellare"), che elimina il carattere precedente la posizione attuale del cursore.

Il programma lascia ampie possibilità di scelta: ciascuno poi finisce per usare il modo di lavoro che gli torna piú comodo. Lavorando regolarmente nella modalità inserimento si corrono forse meno rischi di cancellare qualcosa di importante con un'operazione avventata. Ma gli errori possono capitare sempre. Se ricopiate con un programma di questo genere una lettera, per esempio, e saltate una frase, non avete lo stesso problema della macchina per scrivere: non è necessario, cioè strappare il foglio e riscrivere tutto. Si continua a lavorare sempre sul video (e nella memoria del calcolatore, ma non ce ne accorgiamo), e si può continuare a effettuare correzioni, inserimenti o rifacimenti fino a che non si è soddisfatti, senza toccare le parti che vanno bene.

Una volta soddisfatti del lavoro svolto, è il momento di memorizzare in forma permanente su dischetto il frutto del nostro sforzo. Per memorizzare, passiamo attraverso il menú "blocchi", che si richiama con il comando CONTROL K. Il menú ci offre la possibilità di salvare il nostro documento e di tornare al menú esterno, oppure di continuare nello stesso file, o ancora di uscire dal programma dopo aver memorizzato. Scegliamo di memorizzare e tornare al menú esterno: basta premere D. Il documento dell'esempio è molto breve, basta pochissimo per memorizzarlo e far riapparire il menú esterno. Da questo possiamo procedere all'elaborazione di un altro documento, oppure effettuare qualche altra operazione: per esempio, stampare il documento che abbiamo appena elaborato. Dal menú esterno, basta premere P e il programma si predispone per l'emissione del testo verso una stampante (ovviamente quest'ultima deve essere collegata nel modo corretto al calcolatore e deve essere accesa). Una serie di domande ci permette di fissare alcune particolarità di stampa, ma se non vogliamo cose strane possiamo limitarci a premere il tasto ESCAPE (letteralmente "fuga": è un altro dei tasti con funzioni speciali nella tastiera di un computer) e il programma procederà a eseguire la stampa secondo criteri standard (di cui non dobbiamo preoccuparci minimamente).

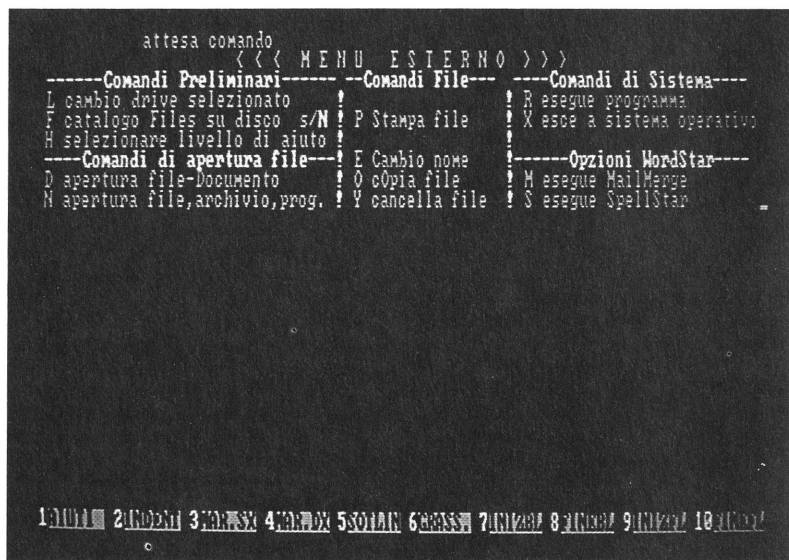
Dopo la lezione

Difficile? No, vero? Quasi tutti i programmi per i personal computer permettono un uso immediato, senza troppe pretese: consentono di cominciare a lavorare senza dover per forza padroneggiare prima tutte le caratteristiche del programma stesso. Poi, con il passare del tempo, acquistando progressivamente confidenza con il programma, si possono esplorare le sue caratteristiche piú avanzate; ma anche conoscendo solo la decina di comandi fondamentali è possibile usarlo proficuamente.

È difficile combinare guai irreparabili, con un personal computer: lo si può ro-

vinare lasciandolo cadere dal tavolo o prendendolo a pedate, ma è pressoché impossibile rovinarlo dandogli dei comandi sbagliati. Può succedere di perdere in tutto o in parte il lavoro fatto, a causa di un comando sbagliato, questo sí: è un'esperienza che bisogna fare almeno una volta, poi si impara a prendere qualche precauzione che non si scorda piú. Può succedere anche di mettere un programma in una condizione di stallo: ci riescono di solito i bambini pestando a caso sui tasti. Ma anche in queste situazioni non c'è alcunché di irrimediabile: in genere è sufficiente spegnere il computer, lasciarlo spento per qualche secondo, riaccenderlo e ricaricare il programma.

Il consiglio fondamentale è dunque: non avere paura! Provare sempre senza timore o soggezione della macchina. Spesso si riesce a usare un programma già la prima volta, senza magari avere il manuale di istruzioni: i menú, qualche indicazione di aiuto spesso sono sufficienti per guidare un utente qualsiasi in una applicazione elementare. Spesso, anzi, leggere il manuale di istruzioni prima di mettere le mani sul programma può addirittura creare piú confusione che chiarezza!

Il programma che ci è servito da esempio (che è il Wordstar della MicroPro International, nella versione 3.30 per il personal computer IBM) è molto piú ricco di quanto abbiamo potuto vedere in queste poche pagine. Tanto per avere un'idea: ci permette di centrare automaticamente titoli; permette di prendere blocchi di testo e spostarli da un punto all'altro del file, oppure di crearne tante copie in punti diversi; permette di concatenare file, o di richiamare un file dall'interno di un altro (caratteristica molto utile per la creazione per esempio di piú lettere che hanno parti in comune: la parte comune è archiviata in un file, il resto delle lettere in altri file, con un richiamo della parte comune nel punto giusto); permette di creare circolari personalizzate, in abbinamento con un altro programma (Mailmerge); e via dicendo.

Il menú di apertura del programma: noi vogliamo scrivere un testo, quindi ci viene suggerito di premere il tasto D per aprire (creare o recuperare) un archivio "documento".

```
d          attesa comando

  Usare questo comando per creare un nuovo documento, o
  per modificare un documento già esistente.

    Un nome di file è lungo 1-8 lettere/numeri e
    opzionalmente "." seguito da 0-3 lettere/numeri.
    Un nome di file può essere preceduto dal nome drive
    A:-D:, altrimenti viene usato il drive selezionato.

^S=annulla carattere  ^Y=annulla parola  ^F=catalogo disco
^D=ripristina car.    ^R=ripristina par. ^U=annulla comando

    NOME DEL FILE DA APRIRE? prova_

1AIUTI 2INDENT 3MAR.SX 4MAR.DX 5SOTLIN 6GRASS. 7INIZBI 8FINEBI 9INIZ3/ 10
```

Premiamo D: il programma ci chiede di dare un nome al file (lo chiameremo "prova"), ci ricorda come deve essere fatto un nome di file e come possiamo effettuare correzioni.

Ora possiamo scrivere: il menú principale ci ricorda le operazioni che possiamo svolgere. Eccoci a fine riga, come si può desumere dal tabulatore: la parola non ci sta tutta.

```
          B:PROVA  PAG. 1 RIGA 2 COL 25            INSERIMENTO
                   < < <  M E N U   P R I N C I P A L E  > > >
   ----Spostamento Cursore--- --Cancella-- -----Comandi Vari----- --Altri Menu--
   CAR:     ^S sx    ^D dx   !^G  car. dx !^I tabulazione        !  ^J aiuti
   PAROLA: ^A sx     ^F dx   !DEL car. sx !^B formatta parag.    !  ^K blocchi
   RIGA:    ^E sopra  ^X sotto!^T parola dx!^V inserimento S/N   !  ^Q rapido
   ---------Scorrimento-------!^Y riga    !^L ripete ricerca/sost!  ^P stampa
   RIGA:    ^Z basso  ^W alto!^U interruz.!RETURN termina parag. !  ^O impost.
   FINESTRA:^C basso  ^R alto!   comando  !^N inserisce RETURN   !
  L----!----!----!----!----!----!----!----!----!----!----!----!--------R
  Quando si arriva a fine riga, però, non c'è bisogno di
  preoccuparsi del margine_

  1AIUTI 2INDENT 3MAR.SX 4MAR.DX 5SOTLIN 6GRASS. 7INIZBL 8FINEBL 9INIZ/ 10FINE/
```

Il programma, verificato che non c'era spazio per tutta la parola sulla riga, l'ha portata automaticamente a capo, spaziando la riga precedente in modo da "giustificarla".

Siamo tornati al menù esterno del programma (vedi pagina 30) e abbiamo premuto il tasto P per stampare. Il programma ci chiede il nome del file che vogliamo stampare.

```
p        attesa comando

NOME DEL FILE DA STAMPARE? prova
ESC per parametri standard di stampa.
  STAMPA SU DISCO? (S/N): _

1AIUTI 2INDENT 3MAR.SX 4MAR.DX 5SOTLIN 6GRASS. 7INIZBL 8FINEBL 9INIZ?/ 10FINE?/
```

Ricevuto il nome del file, il programma pone una serie di domande sulle modalità di stampa. Possiamo aggirarle premendo il tasto "Escape": verranno adottate le modalità standard.

```
        B:CAPIREBI.WSI  PAG. 1 RIGA 3 COL 47        INSERIMENTO
                 < < <  M E N U   P R I N C I P A L E  > > >
 ----Spostamento Cursore--- --Cancella-- -----Comandi Vari----- --Altri Menu--
 CAR:     ^S sx     ^D dx   !^G  car. dx !^I tabulazione        !  ^J aiuti
 PAROLA: ^A sx     ^F dx   !DEL car. sx !^B formatta parag.    !  ^K blocchi
 RIGA:   ^E sopra  ^X sotto!^T parola dx!^V inserimento  S/N   !  ^Q rapido
 --------Scorrimento-------!^Y riga     !^L ripete ricerca/sost!  ^P stampa
 RIGA:    ^Z basso  ^W alto!^U interruz.!RETURN termina parag. !  ^O impost.
 FINESTRA:^C basso  ^R alto!    comando !^N inserisce RETURN   !
 L----!----!----!----!----!----!----!----!----!----!----!-------R
^SI dispositivi di ingresso - il mouse^S                                      <
                                                                              <
La tastiera è uno strumenti dall'impostazione _tradizionale,  con
la  quale molti possono anche trovarsi a proprio agio;  il  mondo
dell'informatica  è  tuttavia molto aperto alla ricerca di  nuovi
strumenti di lavoro, che permettano un modo più semplice e intui-
tivo  di interagire con la macchina. Si possono realizzare  pro-
grammi  che non richiedono sequenze di comandi da  esprimere  per
1AIUTI 2INDENT 3MAR.SX 4MAR.DX 5SOTLIN 6GRASS. 7INIZBL 8FINEBL 9INIZ?/ 10FINE?/
```

Un brano di questo libro, in una fase iniziale di stesura: nella seconda riga c'è un errore. La lineetta prima della "t" è il cursore: qui non si vede, ma sul video lampeggia.

```
        B:CAPIREBI.WSI  PAG. 1 RIGA 3 COL 27
                < < <  M E N U   P R I N C I P A L E  > > >
----Spostamento Cursore--- --Cancella-- -----Comandi Vari----- --Altri Menu--
CAR:      ^S sx     ^D dx   !^G  car. dx !^I tabulazione          !  ^J aiuti
PAROLA: ^A sx       ^F dx   !DEL car. sx !^B formatta parag.      !  ^K blocchi
RIGA:     ^E sopra  ^X sotto!^T parola dx!^V inserimento  S/N     !  ^Q rapido
--------Scorrimento-------!^Y riga      !^L ripete ricerca/sost!  ^P stampa
RIGA:     ^Z basso  ^W alto!^U interruz.!RETURN termina parag. !  ^O impost.
FINESTRA:^C basso   ^R alto!    comando !^N inserisce RETURN   !
L----!----!----!----!----!----!----!----!----!----!----!--------R
^SI dispositivi di ingresso - il mouse^S                               <
                                                                        <
La tastiera è uno strumenti dall'impostazione  tradizionale,  con
la  quale molti possono anche trovarsi a proprio agio;  il  mondo
dell'informatica  è  tuttavia molto aperto alla ricerca di  nuovi
strumenti di lavoro, che permettano un modo più semplice e intui-
tivo  di interagire con la macchina.  Si possono realizzare  pro-
grammi  che non richiedono sequenze di comandi da  esprimere  per
1AIUTI 2INDENT 3MAR.SX 4MAR.DX 5SOTLIN 6GRASS. 7INI28/ 8FINDR/ 9INI29/ 10
```

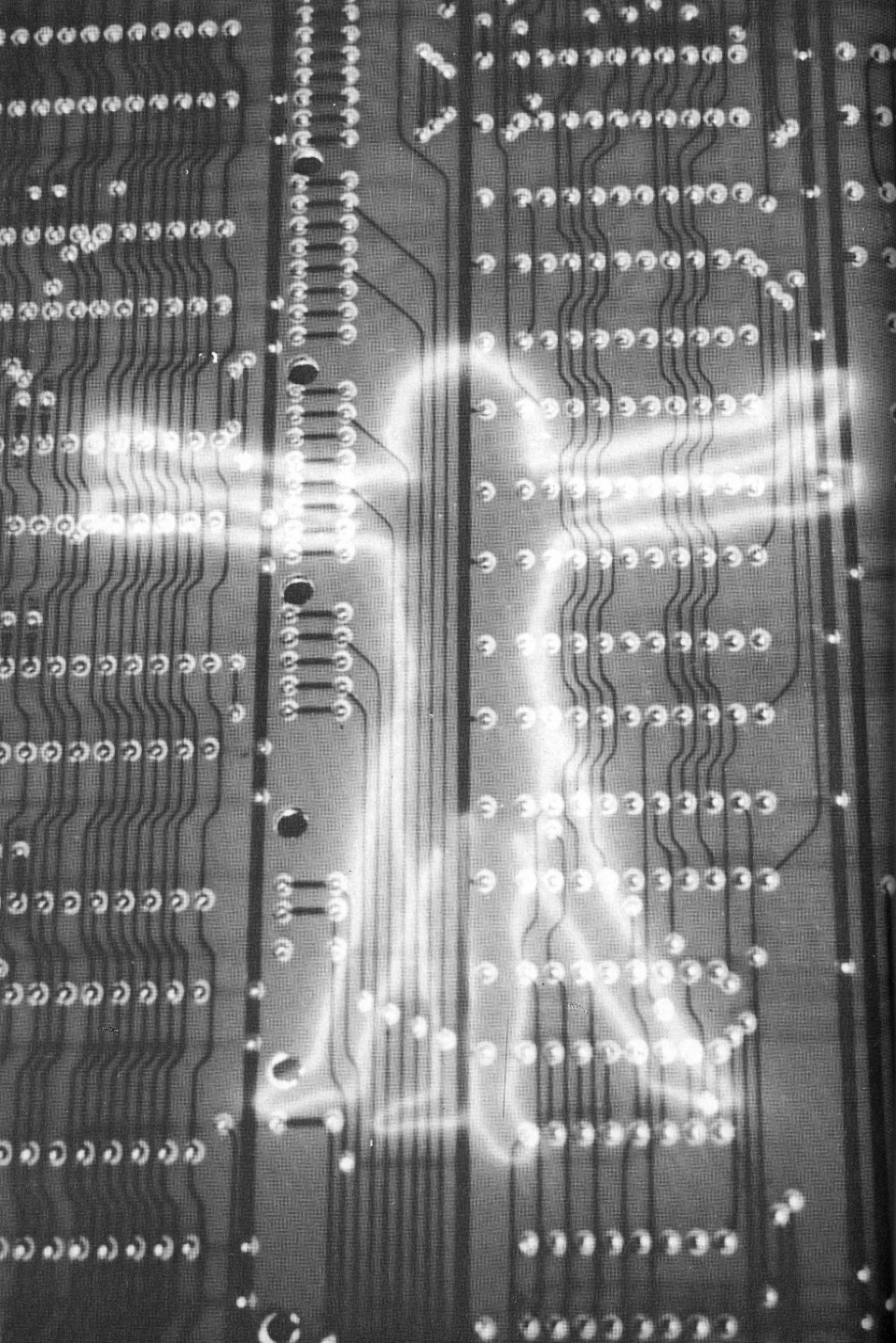
Per correggere basta battere una "o" sopra la "i": per questo, siamo usciti dalla modalità "inserimento" (c'è un comando apposito), che era segnalata dalla scritta in alto a destra.

Corretto l'errore, il cursore si è spostato avanti. Rimanendo in modalità inserimento, la "o" sarebbe stata scritta fra la "t" e la "i", e avremmo dovuto cancellare poi quest'ultima.

```
^O      B:PROVA  PAG. 1 RIGA 1 COL 01                    INSERIMENTO
                  <<<M E N U   I M P O S T A Z I O N E>>>
-Margini e Tabulaz.- --Funzioni Riga-- --Altre Impostazioni--- --Altri Menu--
L def. margine sx   ! C centra testo  ! J giustificazione  n/S!  ^J aiuti
R def. margine dx   ! S def.interlinea! V tab. variabili   n/S!  ^K blocchi
X rilasciare margini!                 ! H aiuto div.parole n/S!  ^Q rapidi
I def. N pulisce tab!---Impostazioni--! E Divisione mobile s/N!  ^P stampa
G tabulaz. paragrafo! W a capo aut.n/S! D visual. comandi  n/S!  ^O imposta
F riga ediz.da testo! I riga ediz. n/S! P salto pagina     n/S!SPAZIO menu pr.
L----!----!----!----!----!----!----!----!----!----!----!--------R

1AIUTO 2INDENT 3MAR.SX 4MAR.DX 5SOTLIN 6GRASS. 7INIZBL 8FINEBL 9INIZIO 10
```

Questo è il menú di impostazione del programma, che permette di definire i margini, attivare e disattivare la giustificazione, centrare un testo, fissare l'interlinea.

```
      B:CAPIREBI.WSI  PAG. 1 RIGA 21 COL 01
L----!----!----!----!----!----!----!----!----!----!----!--------R
^SI dispositivi di ingresso - il mouse^S

La  tastiera è uno strumento dall'impostazione tradizionale,  con
la  quale molti possono anche trovarsi a proprio agio;  il  mondo
dell'informatica  è tuttavia molto aperto alla ricerca  di  nuovi
strumenti di lavoro, che permettano un modo più semplice e intui-
tivo  di interagire con la macchina.  Si possono realizzare  pro-
grammi  che  non richiedono sequenze di comandi da esprimere  per
esteso,  ma a ogni passo offrono all'utente dei ^Smenù^S,  cioè serie
di opzioni alternative,  fra le quali la scelta può essere effet-
tuata premendo semplicemente un tasto.  Integrando questa "tecni-
ca" di programmazione con la grafica,  è possibile rendere ancora
più  semplice l'interfaccia verso l'utente:  con  un  dispositivo
chiamato ^Smouse^S, cioè letteralmente "topolino".

Il  mouse è sostanzialmente un dispositivo di controllo del  cur-
sore:  è  un piccolo oggetto che si può muovere sulla scrivania e
fa spostare sullo schermo il cursore in modo  corrispondente.  Si
sposta  il mouse a destra,  il cursore va a destra;  si sposta il
mouse in avanti, il cursore sale. Il meccanismo di scelta diventa
simile  a quello di certi questionari statistici:  "sbarrare  la
casella corrispondente alla risposta scelta".  Il mouse infatti è
1AIUTO 2INDENT 3MAR.SX 4MAR.DX 5SOTLIN 6GRASS. 7INIZBL 8FINEBL 9INIZIO 10
```

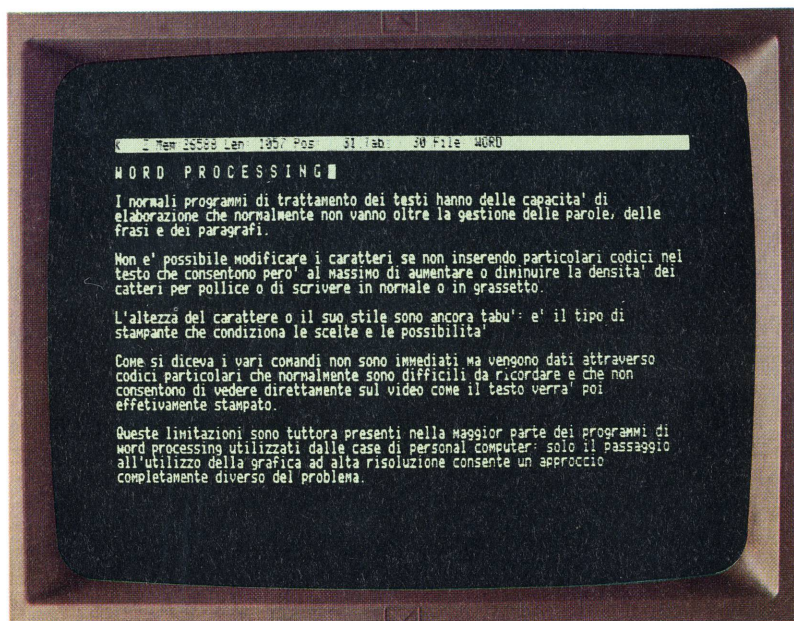
Lo stesso testo di pagina 33 con diversa impostazione: interlinea singola, eliminazione dal video del menú (si vede cosí una maggior quantità di testo con un solo colpo d'occhio).

Il software: ovvero che cosa si può fare

Una lezione di guida non è ancora un corso completo per prendere la patente, ma è già comunque un buon inizio. Speriamo che le pagine precedenti vi abbiano dato un'idea confortante della semplicità d'uso di un personal computer e magari vi abbiano dato un'idea anche della sua utilità. Imparare sul serio a guidare, poi, è piú che altro questione di esercizio, ma per questo è necessario poter disporre effettivamente di una macchina.

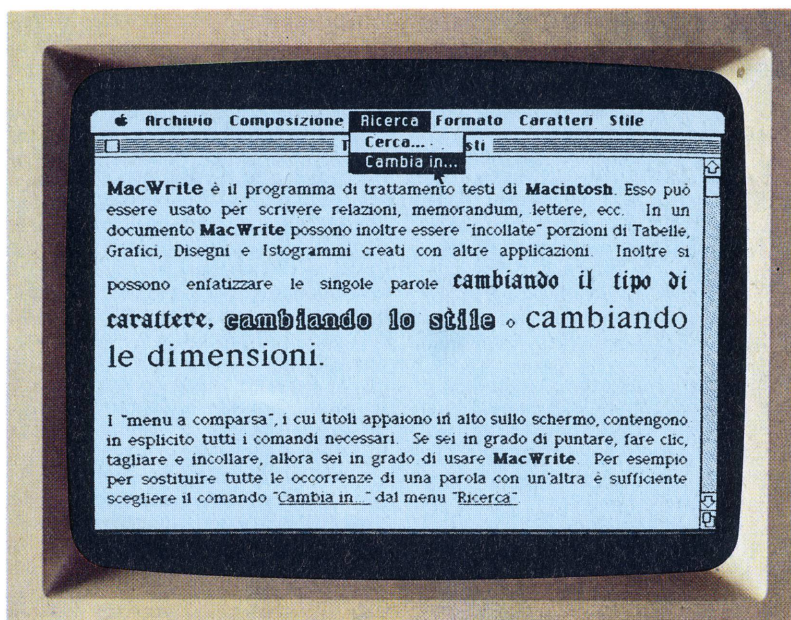
Una buona regola, prima di acquistare un computer, è avere le idee chiare sul motivo per cui lo si vuole, cioè sul tipo di attività in cui lo si vuole utilizzare. Chi acquista un'automobile fa esattamente questo ragionamento: non si acquista una Maserati se si sa di usare l'automobile solamente per spostamenti relativamente piccoli in città: per questo è molto piú adatta una vettura di piccola cilindrata e piccole dimensioni, che abbia consumi contenuti, un costo d'esercizio modesto e semplifichi la vita quando si deve parcheggiare nelle vie della città. Il rappresentante di commercio che deve girare molto e percorrere molti chilometri ogni giorno sceglierà di preferenza una vettura Diesel di cilindrata medio-alta, che gli assicuri affidabilità, comodità e consumi convenienti; chi ha famiglia numerosa vorrà una familiare; e via dicendo.

Nel caso di un personal computer, bisogna avere le idee chiare sugli impieghi che se ne vogliono fare: in altre parole, sul software, sui programmi che si vogliono utilizzare. La scelta dell'hardware, cioè dei componenti fisici della macchina, deve essere secondaria, pena grosse delusioni. Per capire semplicemente come funziona un computer o per poter giocare, qualunque macchina può andar bene, ma per un impiego utile no. Meglio procedere nell'altro senso: cercare di individuare i programmi adatti per l'attività che si deve svolgere, prevedere quali potrebbero essere le espansioni future e poi identificare la macchina o le macchine su cui quei programmi possono girare e verificare se permetteranno in futuro quelle espansioni che ci sembrano utili. Le considerazioni economiche devono venire per ultime: se non potete permettervi la macchina che fa per voi è meglio aspettare per non trovarvi poi amaramente pentiti (costa molto di piú cambiarla in un secondo tempo); oppure si può scendere a un compromesso (ma avendo in mano tutti gli elementi per valutare il costo del compromesso sul buon andamento della propria attività).

In omaggio a questo punto di vista, prima di addentrarci nell'interno del personal computer e di vedere meglio il suo funzionamento e i particolari della sua struttura fisica, dedichiamo le prossime pagine a una rassegna dei tipi principali di software che si possono trovare in commercio, delle loro caratteristiche generali e quindi del tipo di attività per cui possono essere adatti. Non parleremo di programmi specifici: i prodotti esistenti sono troppi e l'evoluzione del

Un computer è tabula rasa, ma ricettiva. I programmi ne determinano le funzioni; per cambiarle basta cambiare il programma.

mercato troppo rapida per rendere sensato un esame minuzioso.

Il word processing

Abbiamo già visto nel capitolo precedente che cosa sia un programma di word processing: è un programma che permette di scrivere testi di qualunque natura, di archiviarli, correggerli, modificarli. È questo uno dei settori in cui maggiore è l'offerta: tutti hanno bisogno di scrivere qualcosa. Esistono però programmi con caratteristiche molto differenziate. I programmi piú semplici permettono l'inserimento del testo, la correzione, operazioni fondamentali di redazione: lo spostamento di blocchi di testo o la loro duplicazione (per riportare brani identici in punti diversi di un documento senza doverli ripetere), la ricerca di stringhe particolari di caratteri e la loro sostituzione in modo automatico con altre stringhe definite sempre dall'utente. Permettono infine di stampare il documento definendo formati di stampa relativamente semplici (numero di righe per pagina, numero di caratteri per riga e simili).
Le funzioni essenziali sono presenti in tutti i programmi, ma i piú raffinati permettono un maggior controllo del formato della stampa, mediante comandi che possono essere inseriti all'interno del testo stesso per modificare, per blocchi particolari, per esempio l'interlineatura, il posizionamento dei margini, la centratura delle righe o la loro giustificazione (una riga si dice giustificata quando riempie esattamente lo spazio fra i margini fissati, come normalmente accade nei libri: la giustificazione si ottiene variando opportunamente gli spazi fra le parole). La maggior parte dei programmi di elaborazione testi non effettua la divisione automatica in sillabe delle parole, per dare una migliore distribuzione del testo fra le righe: i programmi che mettono a disposizione la divisione sillabica sono pochi e fra i piú costosi (è una caratteristica decisamente professionale).

La maggior parte dei programmi non visualizzano il testo come sarà stampato: su alcune macchine orientate alla grafica, come il Macintosh della Apple, si possono visualizzare anche cambi di carattere.

Una differenza essenziale riguarda la presentazione del testo sullo schermo: ci sono programmi che non visualizzano immediatamente come sarà la divisione in righe sullo stampato finale, ma richiedono per questo l'esecuzione di comandi particolari; altri invece sí. Nel primo caso non ci si preoccupa mai dell'aspetto finale del testo nella fase di immissione, ma si rimanda la definizione di tutti i formati alla fine; nel secondo caso è possibile avere almeno alcuni controlli immediati. In un tipo di programma, per esempio, si definiscono i margini all'inizio: la visualizzazione sullo schermo rispetta i margini definiti. Se lo schermo può visualizzare fino a 80 caratteri per riga, ma i margini sono fissati a 10 e 70, per esempio, sullo schermo ogni riga di testo conterrà al massimo 60 caratteri. Anche se il programma non presenta immediatamente la riga giustificata, si può già sapere dove comincia e dove finisce ogni riga, il che può essere utile in taluni documenti dove è necessario disporre il testo in modo particolare. In altri programmi durante l'immissione il testo viene comunque sempre visualizzato sullo schermo a 80 caratteri (massimo) per riga; la definizione dei margini è data solo al momento della stampa e in genere è possibile "previsualizzare" il testo stampato, velocemente, sullo schermo. Però in questo caso si tratta solo di previsualizzazione della stampa: mentre il programma effettua questa operazione, il testo non può essere corretto o modificato. Bisogna uscire dalla funzione di previsualizzazione e rientrare in una funzione di *editing* ("redazione").

Ci sono molte altre differenze di piccola entità: alcuni programmi lavorano sempre implicitamente in modalità "inserimento", altri in modalità "sovrapposizione". Nel primo caso, se ci si posizio-

na con il cursore su un carattere già battuto e si batte un nuovo carattere, questo verrà automaticamente inserito prima del carattere su cui è posizionato il cursore; nel secondo caso il nuovo carattere viene sovrapposto a quello su cui si trova il cursore, cancellandolo. In ambedue i casi è possibile commutare all'altro tipo di funzionamento, quindi non vi è nulla di essenziale; tuttavia qualcuno può trovare piú comodo lavorare con un programma che implicitamente lo metta in una condizione piuttosto che nell'altra. Spesso queste sono caratteristiche da valutare secondo il gusto personale.

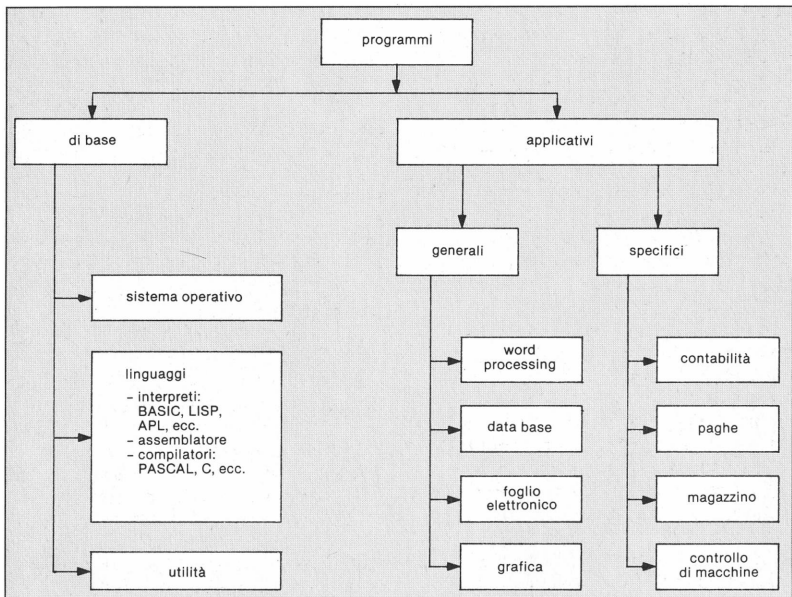
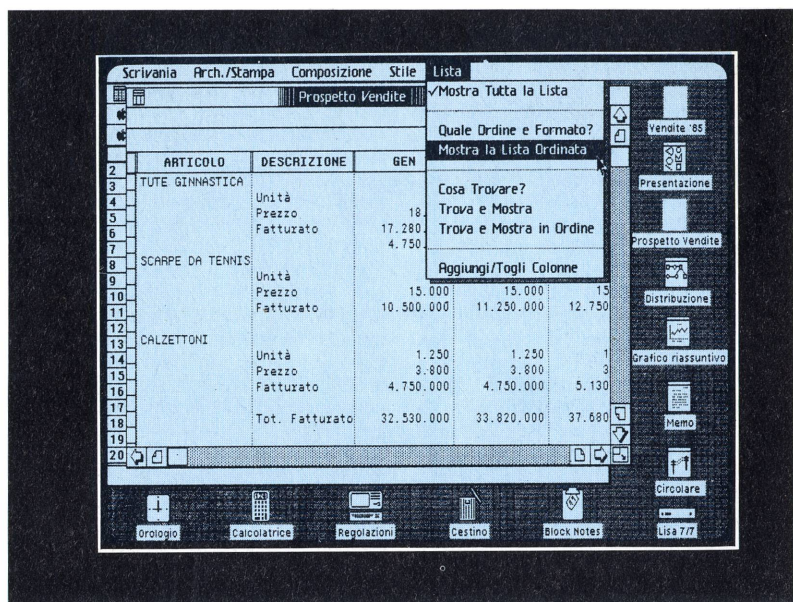
Dove anche i programmi si differenziano è la lunghezza dei testi che possono manipolare (la quale dipende però anche dalle capacità di memoria della macchina). Certi programmi possono trattare al massimo testi di lunghezza predefinita, per esempio 26.000 battute; altri sono piú elastici. Esaurita la quantità di battute permessa dal programma, bisogna memorizzare su supporto esterno il testo che si è scritto, attribuendogli un nome: il file che ne risulta è un'entità autonoma e conclusa. Per procedere a scrivere ulteriormente, si deve iniziare un file diverso: i due file non possono mai essere contemporaneamente nella memoria centrale.

Questo aspetto non è importante se si debbono scrivere solo lettere personali o commerciali, che non hanno mai una lunghezza eccessiva; chi scriva invece articoli, relazioni, libri o traduzioni di una certa lunghezza, può trovarsi a malpartito con programmi che permettono solo file relativamente brevi. Facciamo un esempio. Se ho scritto un libro di duecento pagine, ho scritto qualcosa come 300.000 battute. Il mio programma permette file di lunghezza massima 25.000 battute: dispongo quindi di almeno una dozzina di file, probabilmente di piú (perché ovviamente avrò cercato di spezzare il testo in punti sensati e non matematicamente, magari lasciando spazio per eventuali aggiunte dell'ultimo minuto). Ora che ho finito, scopro di aver sbagliato una cosa, per esempio di aver scritto sempre con l'ortografia errata un certo nome; e quel nome si ripete spesso, per tutto il testo. Non c'è problema, perché dispongo di una funzione di ricerca e sostituzione automatica, che mi permette di dire al programma "cerca il nome scritto cosí e cosí e sostituiscilo sempre con il nome cosí e cosà". Però la funzione agisce solo sul file caricato in memoria centrale: quindi devo caricare un file, attivare la funzione, memorizzare la versione corretta del file, caricare un secondo file, riattivare la funzione, memorizzare la versione corretta di questo file, caricarne un terzo e via dicendo, per tutti i file che ho definito. Caricare un file e memorizzarlo su dischetto sono operazioni che richiedono tempo: tempo variabile a seconda del programma, della macchina, della lunghezza del file stesso, dello spazio libero sul dischetto che ho a disposizione (tutto peggiora quando il supporto di memorizzazione è una cassetta). Una operazione come quella di ricerca e sostituzione attraverso vari file può dunque richiedere parecchio tempo: meglio in questo caso avere pochi file, anche se piú lunghi, perché la funzione di ricerca e sostituzione agisce molto rapidamente e l'elemento critico è dato proprio dalle fasi di caricamento e memorizzazione. Quello della velocità nelle operazioni fondamentali è sempre un aspetto essenziale (non solo per i programmi di elaborazione di testi). È importante, se si pensa di dover lavorare con documenti piuttosto lunghi, che il programma permetta di passare velocemente dall'inizio alla fine del documento, da una videata alla successiva, e via dicendo. È importante anche il tipo di testo che un programma di word processing consente di elaborare: alcuni permettono meglio di altri, per esempio, di scrivere formule matematiche; non tutti i programmi consentono di scrivere esponenti o deponenti (per i numeri di nota, per gli indici di variabili,...); non tutti posizionano automaticamente note; non tutti consentono sottolineature, caratteri diversi, ecc. (Per tutti questi aspetti è indispensabile che anche la stampante disponibile li consenta!)

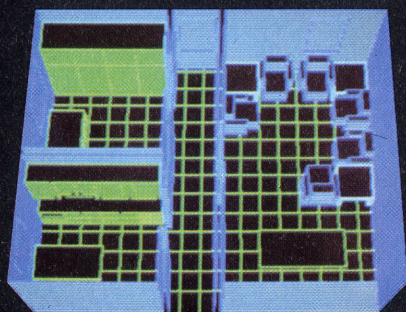
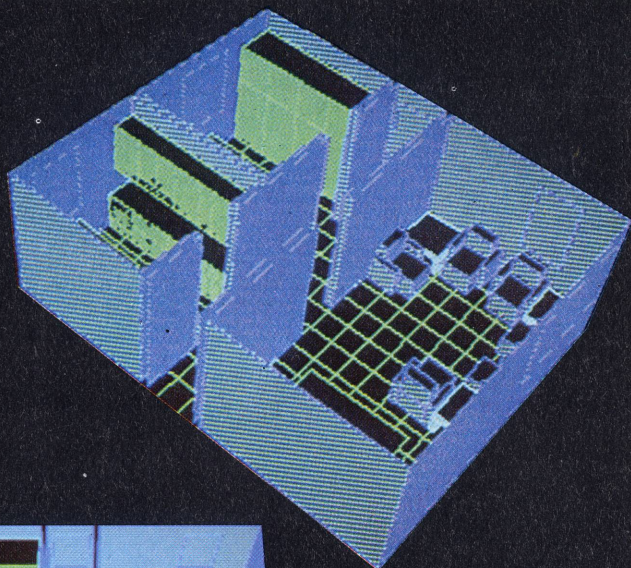
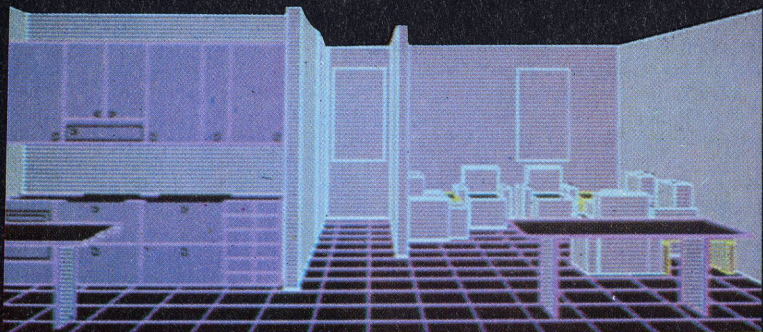
L'integrazione

C'è un'altra cosa da tener presente (ma ne parleremo piú ampiamente verso la fine del capitolo), che vale in particolare per l'elaborazione di testi. Quando si scrive qualcosa, tranne le lettere piú semplici, difficilmente si scrivono parole e basta. Spesso è necessario consultare altro materiale; potrebbe essere utile inserire dati da altri programmi, oppure disegni. Non esistono programmi che facciano tutto: per questo è importante comunque che il programma che si sceglie sia almeno compatibile con altri programmi che si posseggono. Compatibile in questo contesto significa che si deve poter leggere, con il programma di word processing, file realizzati con altri programmi, per poterne inserire parti all'interno di un testo.
Può sembrare banale, ma non è cosí: molti programmi non sono fra loro compatibili, nemmeno in questo senso relativamente debole. Può capitarvi, allora, mentre scrivete una relazione per una riunione importante, di aver bisogno di un istogramma che avete ottenuto il giorno prima con il programma X per l'elaborazione di dati statistici; ma non avete modo di leggerlo direttamente con il vostro programma Y di elaborazione di testi. Cosí siete costretti a lasciare nel documento uno spazio opportuno per l'inserimento dell'istogramma; stampate il testo, separatamente stampate l'istogramma (dopo aver cambiato programma), poi dovete ancora procedere come ai vecchi tempi con forbici e colla. E intanto continuate mestamente a pensare a tutto quello che avete speso per acquistare il computer e i vostri programmi.
Se si prevede qualche volta di aver bisogno di effettuare operazioni simili, è bene pensarci prima e scegliere i propri programmi in modo che sia possibile il passaggio di dati dall'uno all'altro: non sempre è possibile, ma almeno per le applicazioni fondamentali esistono serie di programmi in grado di comunicare

Una classificazione dei programmi che possono girare su un personal computer.

Un esempio di programma applicativo per la gestione di archivi: è una videata da LisaList, programma di archiviazione del personal da ufficio Lisa della Apple. In questo caso la gestione degli archivi è facilitata dal sistema operativo orientato alla grafica, con la capacità di creare finestre e visualizzare contemporaneamente piú documenti diversi.

l'uno con l'altro. Un altro esempio tipico è quello dei programmi per l'archiviazione di dati: può servirvi non solo di memorizzare dati e avere la possibilità di ricercarli secondo chiavi variabili, ma anche di stendere rapporti sulla base di quei dati. In questo caso vi farebbe comodo, prima di procedere alla stampa, rivedere i dati, opportunamente organizzati e selezionati, con un programma che permetta di disporli visivamente nel modo migliore: un programma di word processing sarebbe eccellente, ma può andar bene anche un programma piú limitato, studiato per effettuare operazioni piú semplici solo su archivi di dati.
L'ideale è avere non semplicemente programmi compatibili, ma programmi integrati: che cioè possono essere caricati contemporaneamente e possono interagire direttamente fra loro. In questo caso si può "commutare" rapidamente dall'uno all'altro, senza alcun bisogno di cambiare dischetti, cancellare un programma in memoria e caricarne un altro: si scrive un documento, si va a consultare un archivio di dati, si sposta testo dall'uno all'altro, si consulta un tabellone elettronico, se ne può incorporare una parte senza problemi in un documento... I programmi integrati non sono attualmente molti e sono piuttosto costosi: tuttavia, per usi professionali, si rivelano una soluzione quasi indispensabile.

Archivi di dati

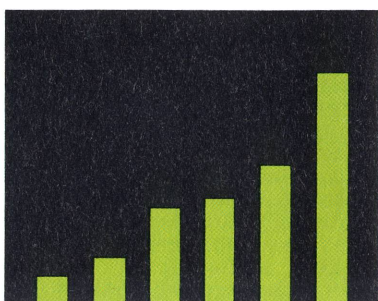
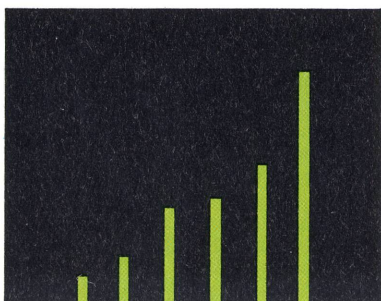
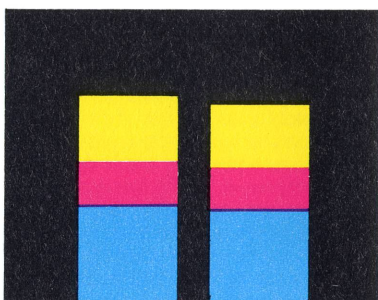
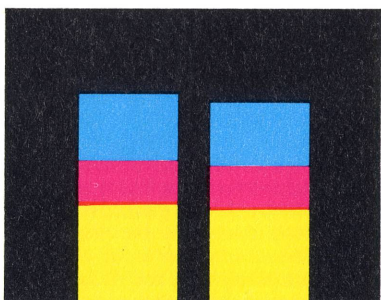
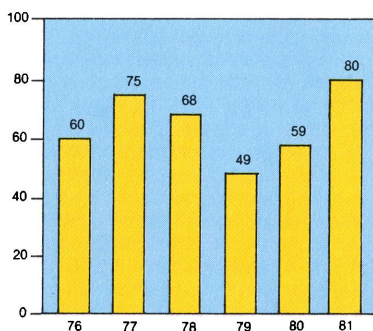
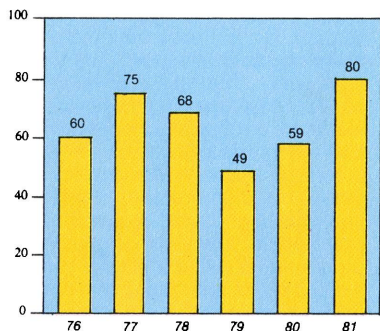
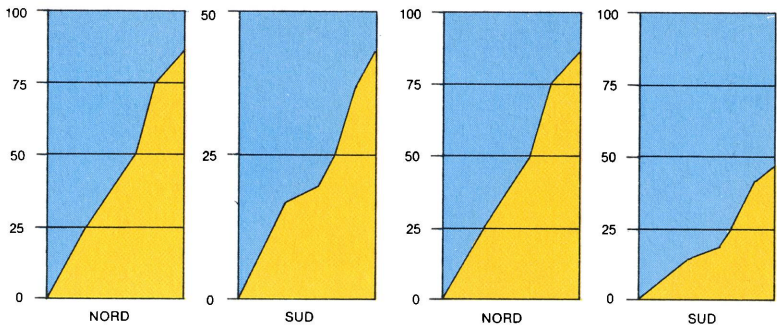
Con il nome inglese di *data base* (base di dati, archivio di dati), sono in commercio numerosi programmi per personal computer che permettono di memorizzare, recuperare, aggiornare ed elaborare informazioni in modo organico. Questi programmi in genere non sono

Un programma Olivetti per l'architettura di interni: nella rappresentazione a tre dimensioni si possono provare comodamente tutte le disposizioni possibili.

predisposti per la gestione di informazioni di natura particolare: sono cioè "pacchetti" di uso generale, che permettono all'utente la creazione di propri archivi entro certi limiti complessivi determinati dalla natura del programma stesso. In altre parole, con lo stesso programma si può creare un archivio bibliografico esattamente come si può tenere un indirizzario o una anagrafica clienti o fornitori: il programma si limita, in un certo senso, a fornire una cornice al cui interno si deve operare. La flessibilità ha in genere un prezzo: questi programmi non rappresentano la perfezione per nessun tipo di archivio. Chi ha esigenze professionali deve ripiegare quasi sempre su programmi *ad hoc*, specificamente studiati per le singole applicazioni e perciò piú efficienti nel loro particolare campo d'azione (ma del tutto inutilizzabili al di fuori). Dipende sempre tutto dalle esigenze: se le esigenze sono limitate, un programma di uso generale ha il pregio di poter essere sfruttato in piú campi; quando le esigenze sono molto forti e molto specifiche (è il caso spesso delle pratiche gestionali delle aziende; non ci sono due aziende che lavorino allo stesso modo, e di solito le differenze non sono varianti immotivate) è piú opportuno ripiegare su prodotti specializzati (ma anche molto piú costosi).
Un programma di gestione d'archivi, generale o specializzato, ha comunque alcune funzioni essenziali che debbono essere presenti:
- deve consentire di memorizzare informazioni;
- deve permettere la ricerca di informazioni, visualizzandole se sono presenti nell'archivio e comunicandone in caso contrario l'assenza;
- deve permettere la modifica delle informazioni memorizzate;
- deve permettere la stampa delle informazioni memorizzate.

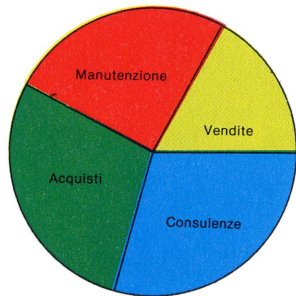
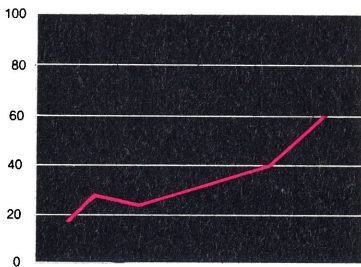
Queste funzioni sono presenti in tutti i programmi: le differenze stanno nel modo in cui vengono realizzate le singole funzioni, e negli elementi ulteriori che ciascun programma può mettere a disposizione dell'utente.
Un programma di data base archivia unità di informazione che vengono chiamate *record*: un record è l'equivalente di una scheda bibliografica, di una scheda anagrafica, o dell'indicazione completa di una voce di inventario. Ciascun record è logicamente suddiviso in *campi*: ciascun campo contiene uno degli elementi che contribuiscono alla costruzione del record.
Un esempio semplice: la costruzione di una rubrica telefonica (o di un indirizzario). Ogni voce della rubrica è un record, e ciascun record può essere costruito in questo modo:
- cognome e nome
- via e numero civico
- città
- codice di avviamento postale
- numero telefonico.

Ciascuno di questi elementi costituisce un campo del record. Un programma specializzato per la gestione di una rubrica telefonica con indirizzario ce li fornirebbe già pronti (magari in un ordine diverso, magari con altri elementi), mentre nel caso di un programma di tipo generale la definizione dei campi è lasciata all'utente.
Il modo di definire record e campi varia da programma a programma: vi sono programmi che fissano limiti di lunghezza al record o ai singoli campi, programmi che richiedono di specificare se i contenuti di ciascun campo sono caratteri o numeri, programmi che organizzano record e campi in forma di tabelle e altri che li organizzano a schede... Non esistono regole generali. Una volta fissata la struttura dell'archivio, il programma permette l'immissione dei dati, visualizzando volta a volta la struttura del record e consentendo all'utente di riempire con i contenuti opportuni i vari campi. Quasi sempre in questa fase il procedimento non è dissimile da quello normalmente impiegato per riempire un questionario o un modulo in cui le informazioni da comunicare sono segnalate da file di puntini, tipo:

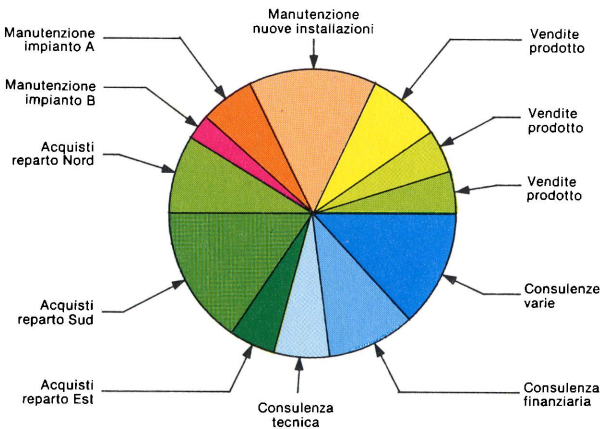
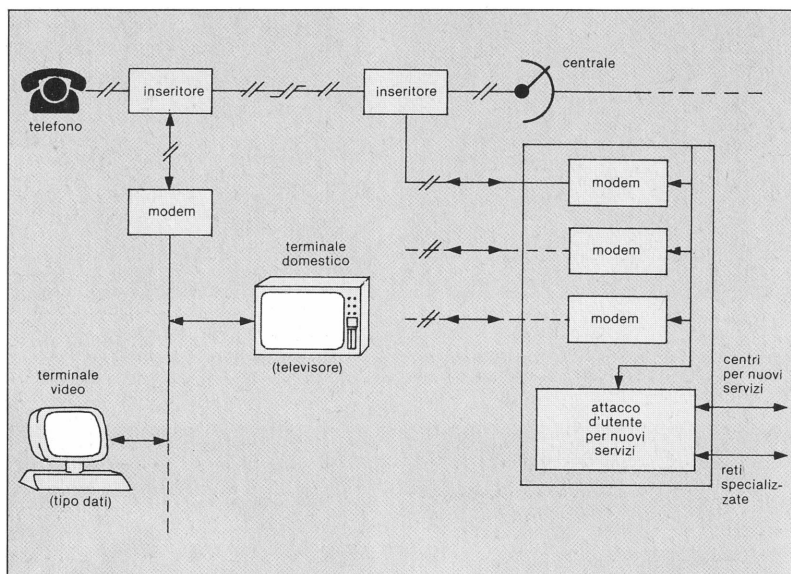
Il sottoscritto .... nato a ....

(prov. ......) il ....

Nei programmi che pongono limitazioni alla lunghezza di campi e record, nella fase di immissione dei dati è ovviamente indispensabile a volte fare un po' di attenzione, per rendere il piú possibile compatto il materiale e non superare i limiti.
L'archivio viene conservato, poi, su dischetto (o su cassetta, per i sistemi piú economici): la grandezza dell'archivio è a quel punto funzione della capacità del dischetto e della lunghezza dei singoli record. Se un dischetto, poniamo, può contenere al massimo circa 120.000 caratteri e ciascun record è lungo 200 caratteri, sul dischetto potranno trovar posto al piú 600 record. Il programma può comunque "rubare" un po' di posto, nel dischetto, per depositare informazioni utili alla propria attività, in fase di ricerca e di recupero, quindi non sempre il calcolo è preciso; resta tuttavia indicativo delle dimensioni possibili dell'archivio.
600 record sono troppo pochi per il vostro archivio? Dovete sistemarlo su piú dischetti, se è possibile logicamente senza incappare in una scomodità eccessiva; oppure dovete scegliere un computer che vi permetta di usare dischetti piú capienti; oppure dovete arricchire il vostro hardware di un disco rigido (i dischi rigidi hanno capacità di qualche milione di caratteri, fino anche a 20-30 milioni e piú).
Nel caso di un programma di data base, le dimensioni della memoria centrale dell'unità di elaborazione costituiscono un limite alle prestazioni in termini di qualità e velocità: un programma piú complesso e piú veloce richiede in genere una maggiore memoria centrale. Le dimensioni possibili dell'archivio sono invece limitate dalla capacità della memoria di massa esterna. Per gestire una agenda telefonica che contiene solo nomi e numeri telefonici non c'è bisogno di un programma molto complicato: la memoria centrale può anche essere piccola. Ma se l'archivio deve avere le dimensioni delle "pagine gialle" di una grande città, qualunque sistema di memoria di massa a dischetti flessibili sarà inadeguato. Le piccole dimensioni della memoria centrale possono avere un'influenza, in un caso di questo genere, sulla velocità di esecuzione del programma: all'interno della memoria principale le operazioni sono velocissime, mentre tutte le operazioni di accesso al disco (in generale a qualunque unità esterna) sono al confronto molto lente. Se il programma deve accedere continuamente al disco per ricercare un'informazione, il suo funzionamento sarà lento; con una memoria centrale di maggiori dimensioni può invece caricare contemporaneamente quantità maggiori di dati ed effettuare ricerche piú veloci.
Gli elementi da tenere in considerazione nella scelta sono dunque numerosi, ma dobbiamo aggiungere qualche altro fattore. Non tutti i progammi memorizzano le informazioni nello stesso modo: alcuni, piú raffinati, le codificano in forme, per cosí dire, stenografiche; l'utente non se ne avvede, ma in questo modo il programma può gestire con maggiore efficienza la memoria di massa. Si memorizzano informazioni per poterle poi ritrovare: è questo il punto nodale. I programmi di data base permettono di ricercare informazioni sulla base della struttura che è stata loro data. Nel caso della rubrica telefonica, per esempio, potremo ricercare le informazioni che si riferiscono al signor Rossi Mario perché abbiamo predefinito un campo "cognome e nome". La ricerca avviene di norma attraverso i campi di definizione: in qualche forma il programma chiede attraverso quale campo si vuole effettuare la ricerca, per esempio ripresentando la struttura dei record e chiedendo di identificare il campo di ricerca. Un po' come in certi moduli: "sbarrare la casella relativa alla voce che interessa". Sui moduli c'è quasi sempre lo spazio per una voce non contemplata da chi ha predisposto il modulo stesso, ma nei programmi di data base no. Se non si è definito un campo "città" non si può andare a ricercare, fra tutti i numeri di telefono archiviati, quelli di Milano.
Se l'informazione non è stata mai archiviata, ovviamente, non c'è nessun modo di recuperarla: il computer non si so-

Una immagine significativa val piú di molte parole: specialmente tra i manager piú impegnati la business graphics, la grafica commerciale, ha una grande diffusione. Ma attenzione ai particolari! In un istogramma, per esempio, barre strette e spazi grandi fanno perdere di vista la cosa piú importante.

Nella grafica a colori, la scelta delle tonalità risulta di estrema importanza; una regola semplice ma molto efficace richiede che nella colorazione di un istogramma i colori scuri vengano collocati piú in basso, quelli chiari in alto.

Nella grafica la coerenza è importante anche nei minimi particolari: non si devono creare differenziazioni inutili. Nel diagramma a sinistra, per esempio, l'uso di tondo, corsivo per i numeri non ha nessun senso: meglio un unico stile, come a destra.

Ancora una questione di uniformità che diventa significativa: quando si mettono a confronto grafici diversi, correttezza richiede che le due rappresentazioni siano alla stessa scala. Altrimenti tutta l'efficacia del confronto visivo immediato va perduta.

Manutenzione
Vendite
Acquisti
Consulenze

In un grafico, la curva che riporta il valore da esaminare deve essere piú evidente del sistema di riferimento.

In un grafico a torta, poi, meglio un po' di aggregazione dei dati che un dettaglio, corretto sí, ma illeggibile.

Una possibilità del futuro non troppo lontano: la filoinformazione, una tecnica che permette di trasmettere nello stesso tempo, su una normale linea telefonica, le conversazioni vocali e dati digitali per le comunicazioni tra computer.

gna cose che non gli diciamo espressamente. Se per esempio avessimo memorizzato il nome della città insieme alla via e al numero civico, quasi tutti i programmi ci permetterebbero in qualche modo di risalire all'informazione: si può quasi sempre effettuare ricerche per voci "incomplete". Cioè si può chiedere al programma, non solo "ricercami tutti i record che nel campo tal dei tali hanno il nome Milano", ma anche "ricercami tutti i record che nel campo tal dei tali hanno il nome Milano, eventualmente in mezzo ad altre cose". Se il campo "tal dei tali" è quello dell'indirizzo, in questo modo potremmo recuperare tutti i numeri telefonici di Milano, ma non solo quelli: anche i numeri telefonici di persone che abitano in via Milano, in qualunque paese o città.
Il grado di precisione delle risposte del programma è sempre funzione del grado di precisione della richiesta. Quasi tutti i programmi di data base permettono di effettuare *ricerche incrociate*, cioè attraverso piú *chiavi* di ricerca. Per continuare nell'esempio, potremmo chiedere una ricerca di tutti i numeri telefonici di Milano, poniamo, che corrispondono alla zona con codice di avviamento postale 20131.
I campi dei record definiscono, dunque, le chiavi di ricerca: senza chiavi non si apre nessuna porta. Ciascun programma, poi, differisce dagli altri per la complessità degli "incroci" che permette fra le chiavi. Le possibilità sono molte, perché si potrebbe desiderare di effettuare ricerche:
– su tutti i record che contengono la chiave A *e* la chiave B (o piú di due);
– su tutti i record che contengono la chiave A *o* la chiave B;
– su tutti i record che *non* contengono la chiave A;
– su tutti i record che contengono la chiave A *e non* contengono la chiave B;
e le varie combinazioni di *e*, *o* e *non* che si possono costruire. Ma non finisce

qui. Supponiamo di avere un archivio fatture. Potrà interessarmi anche sapere tutte le fatture emesse *dopo* un certa data, oppure *prima* di una certa data, o ancora *fra* due date ben precise. Debbo avere un campo "data di emissione", ma soprattutto debbo avere la possibilità, in fase di ricerca, di usare degli *operatori relazionali*, come "maggiore", "minore" o "compreso fra". Lo stesso vale nel caso si vogliano sapere tutte le fatture per un importo superiore a 10 milioni, e via dicendo.

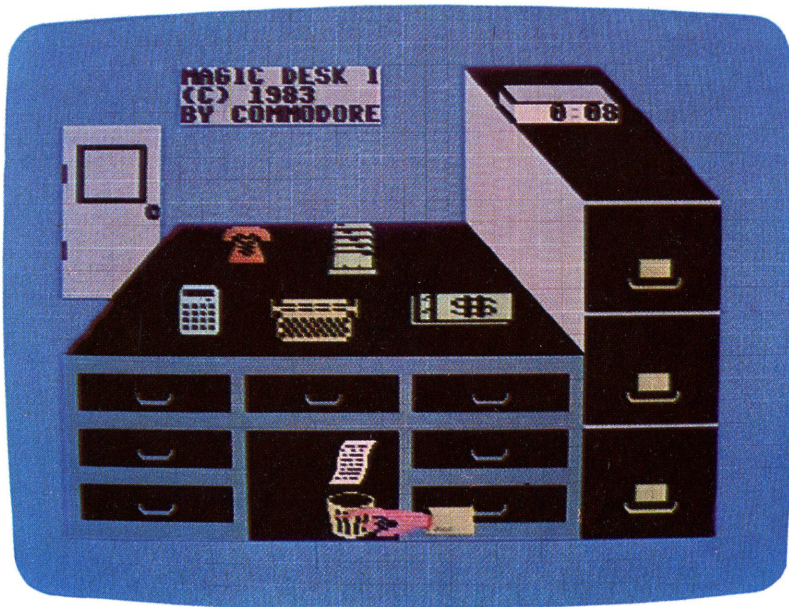
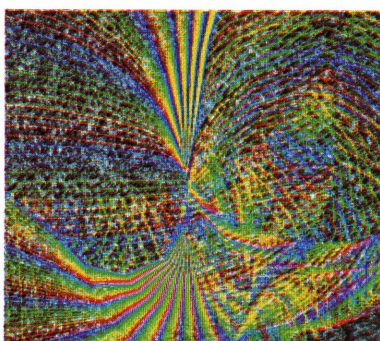
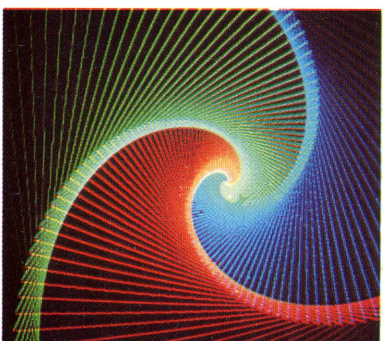
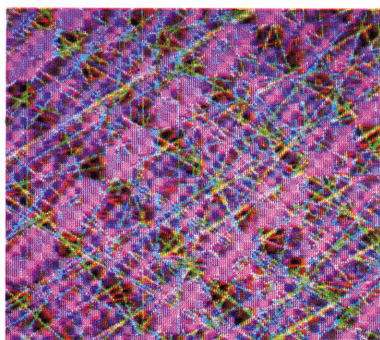
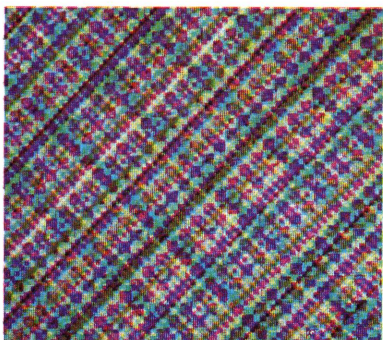
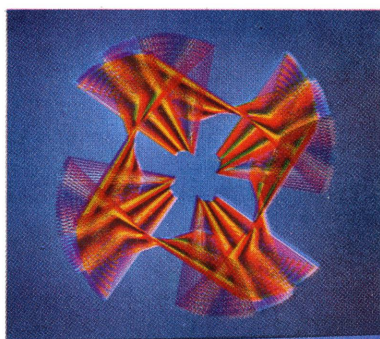
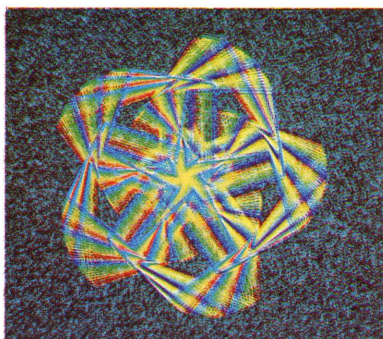
Esistono anche programmi di data base *a schema libero*, che mettono a disposizione l'equivalente di una schedina di carta, rappresentata da una schermata completa: questi programmi permettono sia di trattare la schermata come un record piú convenzionale e di inserire indicazioni di campi, sia di trattarla come un'unità complessiva non articolata in campi. In questo caso, in genere è possibile usare come chiave di ricerca qualunque parola possa figurare nelle schede. Inevitabilmente, un programma che presenti queste caratteristiche sarà piú complesso e piú lento di un programma che possa lavorare su campi predefiniti e quindi abbia la strada in qualche modo già tracciata, ma è altrettanto vero che permette una maggiore libertà nel tipo di informazione archiviabile. Programmi del genere sono utili soprattutto per archiviare appunti sparsi – quelli che normalmente annoteremmo su un foglietto volante – di qualunque genere e tipo, senza bisogno di classificarli in alcun modo. Poi è sufficiente ricordare che si era preso un appunto su una mostra di cani a Monza per recuperare l'informazione: "mostra", "cani" e "Monza" sono tre chiavi che probabilmente se non identificano esattamente l'appunto, almeno restringono enormemente le possibilità. Il programma ci presenterà in successione tutte le schedine che contengono tutte e tre quelle parole (o, impostando in altro modo la ricerca, almeno una delle tre: potremmo anche non ricordare se abbiamo scritto tutte quelle cose, in origine). Potremo cosí sapere la data della manifestazione, il luogo esatto e tutte le informazioni che avevamo memorizzato originariamente.

In che ordine, eventualmente, un programma ci fornisce i vari record, che soddisfano i requisiti imposti nella nostra richiesta? E in che forma? Anche qui le possibilità sono molte. Ciascun programma "predilige" una forma particolare. Le alternative principali sono due: un record per volta oppure piú record insieme, in forma di tabella; in ordine alfabetico rispetto a una chiave di ricerca specificata, oppure nell'ordine in cui sono stati memorizzati (o nell'ordine contrario, a partire da quelli di immissione piú recente). Ciascun modo di presentazione ha i propri pregi e i propri svantaggi, che vanno commisurati volta a volta alle specifiche applicazioni. Ci sono anche programmi che permettono di ricercare secondo certe chiavi di ricerca e di ordinare i risultati, per esempio in ordine alfabetico, rispetto a una voce diversa.

Tutte queste caratteristiche sono importanti quando si vogliono creare, da un archivio principale, archivi secondari in cui determinate informazioni siano già organizzate nel modo piú opportuno per compiti particolari, e quando poi si vogliono i risultati a stampa. Esistono peraltro programmi per permettere di gestire dettagliatamente anche la stampa dei risultati, in modo da generare "*report*", relazioni, specchietti riassuntivi e simili, secondo direttive che possono variare di volta in volta.

Altri programmi offrono poche scelte sotto questo profilo: sono pensati sostanzialmente per l'uso su video. Altri ancora non hanno caratteristiche pregevoli per la stampa, ma possono essere usati in abbinamento ad altri programmi che, sfruttando gli stessi dati, sovraintendono specificamente alla realizzazione di report. Chi pensa di usare solo il video può accontentarsi del programma principale, che comunque è piú che sufficiente per una stampa occasionale; chi invece sa di dover realizzare frequentemente comunicazioni a stampa sui propri dati può acquistare anche l'altro programma: è una soluzione commerciale interessante.

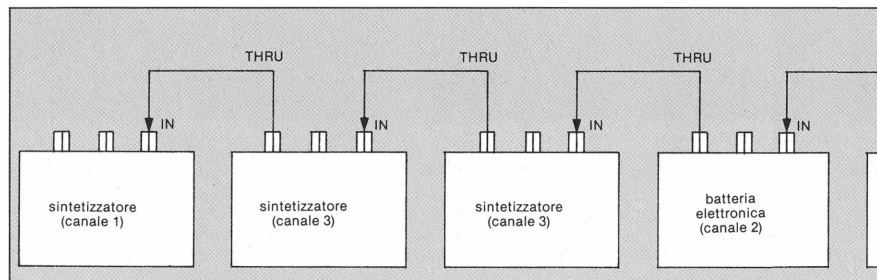
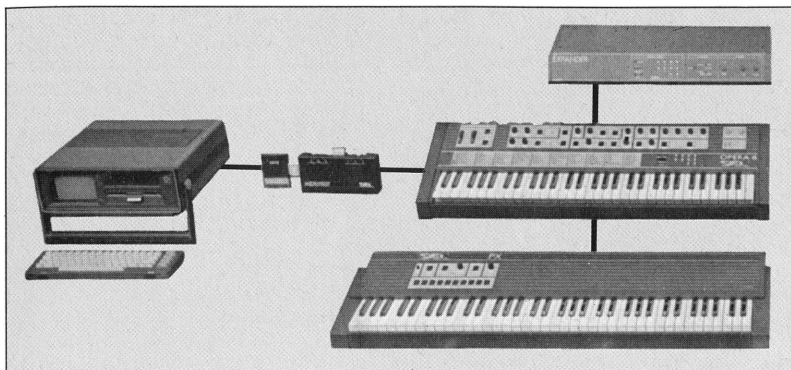
Sistemi di archiviazione e DBMS

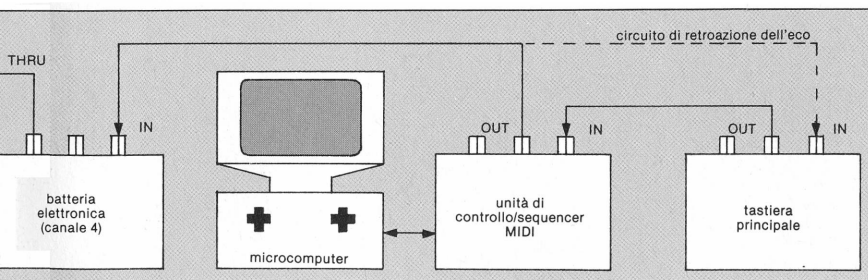
I programmi di cui abbiamo parlato finora sono quelli della fascia bassa e media: adatti per l'uso domestico o per un uso professionale non troppo spinto. Con terminologia anglosassone, sarebbero *file system* o *file manager*, sistemi o gestori di archivi. I programmi della fascia superiore, i piú raffinati, vanno invece sotto il nome di *Data Base Management System*, o in sigla DBMS (sistemi di gestione di basi di dati), secondo una terminologia che ha avuto origine nel mondo dei calcolatori di grandi dimensioni. Un vero DBMS non si limita a offrire quelle capacità di archiviazione, recupero e ordinamento delle informazioni che abbiamo visto fin qui: offre un vero e proprio linguaggio di programmazione per la gestione delle informazioni. Si possono programmare sequenze di comandi, anche molto lunghe e complesse, che poi vengono memorizzate esattamente come qualunque altro programma: in seguito è sufficiente richiamare il programma e mandarlo in esecuzione perché il sistema effettui una serie notevole di operazioni. Questa caratteristica è molto utile a tutti coloro che, per esempio, periodicamente debbono presentare rendiconti e quadri riassuntivi della situazione (tipicamente di un'azienda) e per questo debbono poter elaborare grandi quantità di dati. L'elaborazione può contare un gran numero di fasi, ma ogni volta (ogni mese, ogni trimestre...) si tratta sempre di quella serie di fasi, quindi programmabili una volta per tutte.

Un programma semplice, per un personal computer di tipo domestico (si tratta del diffuso Commodore 64), in cui tuttavia si rivelano già operanti alcuni criteri di fondo del software piú evoluto nell'informatica personale. Il Magic Desk è un programma con un'interfaccia grafica molto intuitiva: usa la metafora della scrivania e degli oggetti che normalmente ne costituiscono il corredo e il contorno, per facilitare all'utente l'interazione nella gestione dei suoi archivi. Il simbolo della mano, spostato con il cursore, indica l'operazione che si vuole eseguire.

Un esempio di computer art, firmato da K. Kahn e collaboratori dell'Artificial Intelligence Laboratory del Massachusetts Institute of Technology. Sono fotogrammi di un film d'animazione ottenuto al computer (in questo caso specifico un "mini" della Digital). Simili film astratti raffiguranti forme e schemi geometrici in movimento sono nati dalla collaborazione fra artisti e informatici: anche per i personal computer esistono ormai programmi di grafica molto raffinati, che permettono di ottenere immagini di qualità non molto inferiore.

Usare un DBMS completo non è facile: offre sempre un numero molto elevato di funzioni, di gran lunga superiore a quello di un comune sistema di archiviazione personale, e richiede, per un uso sensato, una notevole quantità di tempo dedicata allo studio. In genere non è il tipo di programma piú consigliabile per un principiante, ma per risolvere problemi di natura professionale può essere l'unica soluzione valida.

Compatibilità dei dati

Le informazioni che si desidera archiviare in genere non sono statiche e difficilmente sono utili di per se stesse, cosí come sono. In altre parole, variano nel tempo (quantitativamente e magari anche qualitativamente) e debbono essere utilizzate per fare altre cose (scrivere libri, preparare relazioni o stilare la classifica dei capocannonieri del campionato di calcio, poco importa). L'osservazione non è banale: quando si imposta una base di dati, bisogna sempre pensare al futuro. Ci sono non solo problemi di hardware (il supporto di memoria che uso può diventare troppo limitante perché permette di archiviare un numero troppo piccolo di informazioni), ma anche di software: la struttura che oggi impartisco ai dati, con l'aggiungersi delle informazioni, fra sei mesi, forse un anno o due, potrebbe non andarmi piú. Che cosa fa il mio programma? Mi permette di modificare la struttura salvando tutti i dati? O mi costringe a ripartire dall'inizio con la nuova impostazione, magari a reinserire tutti i dati manualmente secondo il nuovo schema?

Se il cambiamento è molto grande, ci sarà ben poco da fare: bisognerà ripartire da zero. Ma a volte si tratta solo di aggiungere un campo o due, di toglierne un altro. Per restare a livello di applicazioni domestiche: potrei aver realizzato una base di dati con tutti i dischi che possiedo, classificandoli per autore, esecutore, titoli, sigla, genere musicale e simili. Dopo sei mesi potrei scoprire che mi farebbe comodo avere un campo in piú, in cui di volta in volta segnare il nome della persona a cui ho prestato quel disco, per potermelo ricordare. L'ideale è poter fare come con le schedine tradizionali: al fondo di ogni scheda aggiungo una annotazione. In termini di programma di archiviazione: conservo il tutto, mi limito ad aggiungere alla struttura un nuovo campo. Non tutti i programmi lo consentono, anche se può sembrare una cosa banale!
Il fatto che le informazioni non siano interessanti di per se stesse, ma servano per scopi variabili di volta in volta, significa che può essere interessante estrarre dati da un archivio e farli elaborare opportunamente ad altri programmi, magari per ricavarne analisi statistiche, o visualizzazioni grafiche in forma di istogrammi, diagrammi a torta e simili. Beh, anche questa non è un'operazione scontata. Anzi, nella maggior parte dei casi è un'operazione impossibile. Ogni programma ha il suo modo di scrivere e leggere dati – parla in un certo senso una propria lingua – e un programma non è detto possa leggere ciò che è stato scritto da un altro programma. Può sembrare assurdo, ma le cose stanno proprio cosí!

Molti strumenti musicali (sintetizzatori, batterie elettroniche, sequencer, ecc.) sono costruiti con tecnologia digitale e sono controllati da un microprocessore: è pensabile metterli in comunicazione con un personal computer e, da questo, impartire istruzioni di controllo a piú strumenti contemporaneamente, per l'elaborazione di suoni o per l'esecuzione coordinata di brani musicali. Oltre alla disponibilità del software adeguato, il problema è dato soprattutto dalla necessità di porre fisicamente in comunicazione i vari tipi di dispositivi. Nel 1984 ha cominciato ad avere diffusione commerciale una interfaccia studiata appositamente a tale fine, che prende il nome di MIDI (Musical Instruments Digital Interface), che fissa uno standard per il trasferimento di segnali digitali fra macchine diverse. Uno strumento dotato di questa interfaccia presenta una presa IN di ingresso, una OUT di uscita dei segnali, e spesso anche una presa THRU che permette il passaggio senza elaborazione (per esempio per il controllo in serie di piú macchine). L'interazione è possibile "in tempo reale".

Questa situazione non ha motivazioni reali: è piuttosto il portato di una evoluzione troppo veloce e disordinata, senza alcuna forma di controllo. Ogni produttore ha realizzato i propri programmi senza tener conto di quelli preparati da altri, o magari addirittura tenendone conto, ma di proposito cercando di differenziarsi per motivi di concorrenza. A volte anche all'interno della produzione di una medesima ditta non c'è compatibilità! Non esistono veri standard a cui tutti si adeguino: è un dato generale per tutto il mondo dei personal computer che non ha nulla a che fare con la natura di queste macchine ma ha radici puramente commerciali; grazie al progressivo smaliziarsi degli utenti, oggi comincia a essere corretto. Se di standard ancora non si può veramente parlare, tuttavia i pacchetti di software coordinato o integrato (vedi a pagina 67) svolgono un'azione salutare. Nel frattempo: attenzione agli acquisti!

I tabelloni elettronici

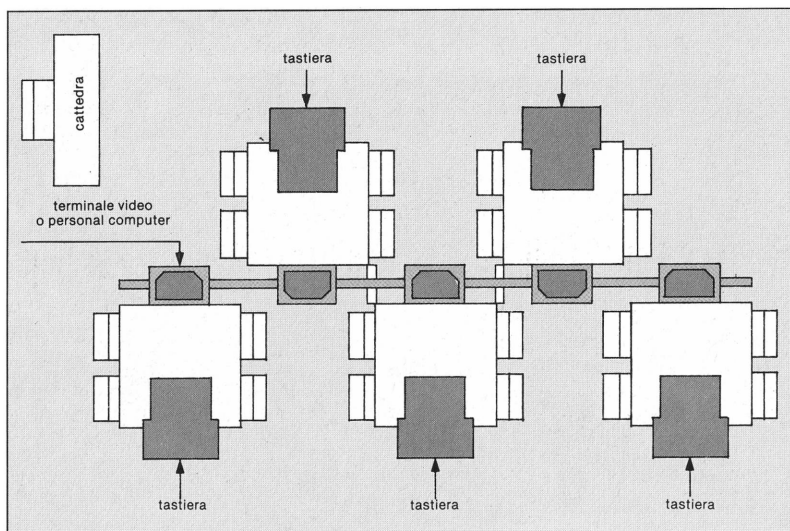
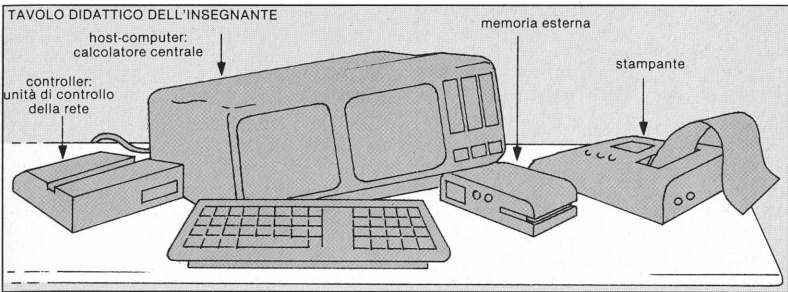
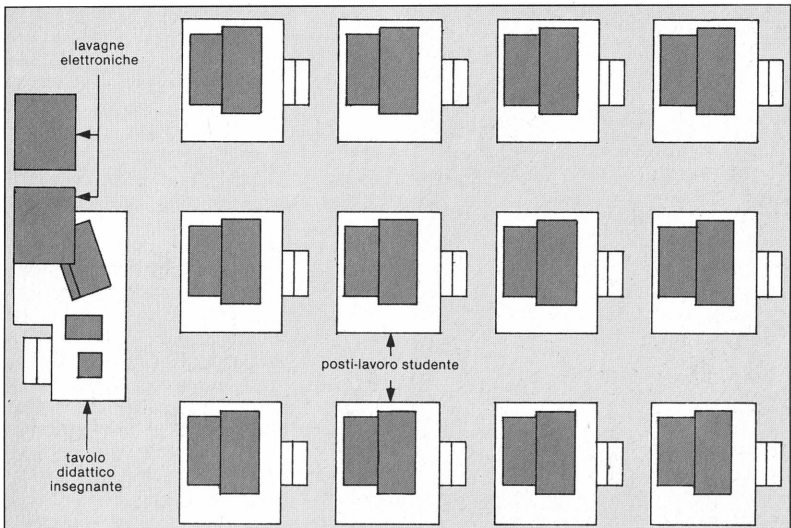
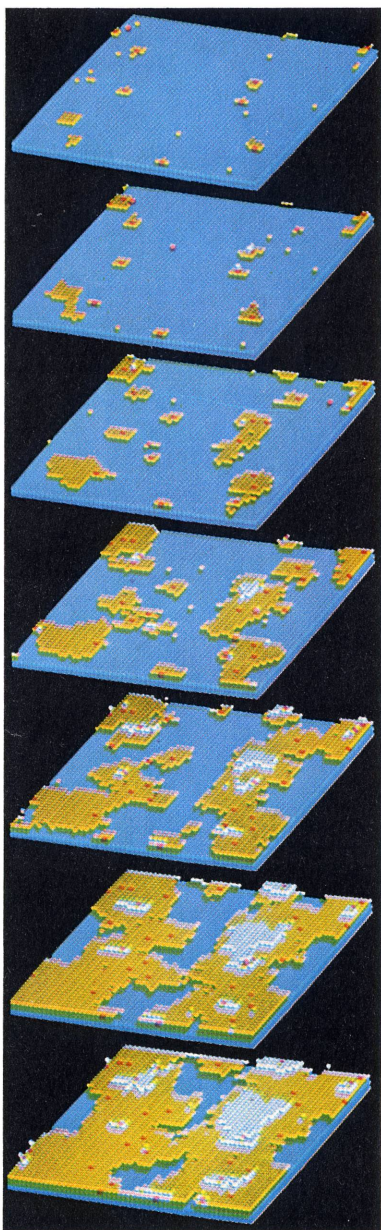
Tabellone, tabella, foglio elettronico: non esiste una traduzione universalmente accettata per il termine inglese *spreadsheet*, che indica un tipo di programma particolare, fra i piú utili e i piú caratteristici creati espressamente per il mondo dell'informatica personale. La struttura generale di questi programmi è quella di un tabellone organizzato per righe e colonne, proprio come una grossa tabella o una matrice, in modo da creare una schiera di caselle. Lo si usa piú o meno come un tabellone di carta: in ogni casella si possono inserire dati particolari, per esempio sulla prima riga in corrispondenza di ciascuna colonna i dati di vendita di un determinato prodotto, mese per mese. Analogamente si possono riportare i dati di vendita di un secondo prodotto, sempre mese per mese, sulla seconda riga, e via dicendo. Fin qui nulla di interessante. Il bello è la possibilità di attribuire alle caselle anche non dati, ma formule. Per comodità, possiamo identificare le righe con una lettera dell'alfabeto e le colonne con un numero: A1 sarà la prima casella in alto a sinistra, all'incrocio della prima riga con la prima colonna; D16 sarà la casella all'incrocio della sedicesima colonna con la riga D (la quarta). Molti programmi di tabellone elettronico usano proprio questa convenzione. Allora, potremmo anche attribuire alla casella A13 la formula (A1 + A2 + ... + A12), cioè "calcola la somma delle prime dodici caselle su questa stessa riga". Alle dodici caselle precedenti avevamo depositato i dati relativi alle vendite mensili di un certo prodotto: in questo modo la tredicesima ci darà il totale annuale. La cosa importante è che nella casella A13 non andiamo a mettere un dato, ma una formula: è il programma che si preoccupa di calcolare il valore da inserirvi. La formula rimane sempre associata alla casella: se si modifica un dato in una delle caselle da cui dipende il valore di

questa, il programma è in grado di effettuare automaticamente un nuovo calcolo, riaggiornando tutti i valori.
Il primo tabellone elettronico è stato il Visicalc, che ha fatto la fortuna della Visicorp Co., e rimane un prodotto valido e interessante a distanza di anni dalla sua creazione (è stato aggiornato regolarmente): hanno fatto seguito decine di programmi basati sullo stesso impianto, ma variamente differenziati per caratteristiche come il numero massimo di ca-

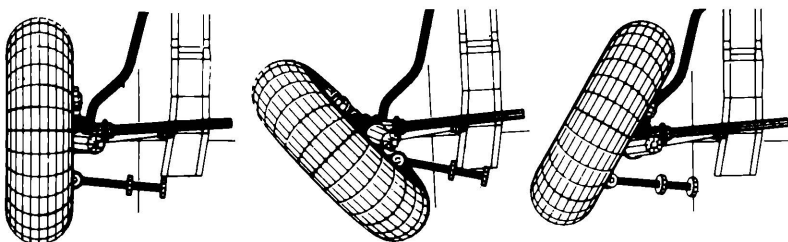
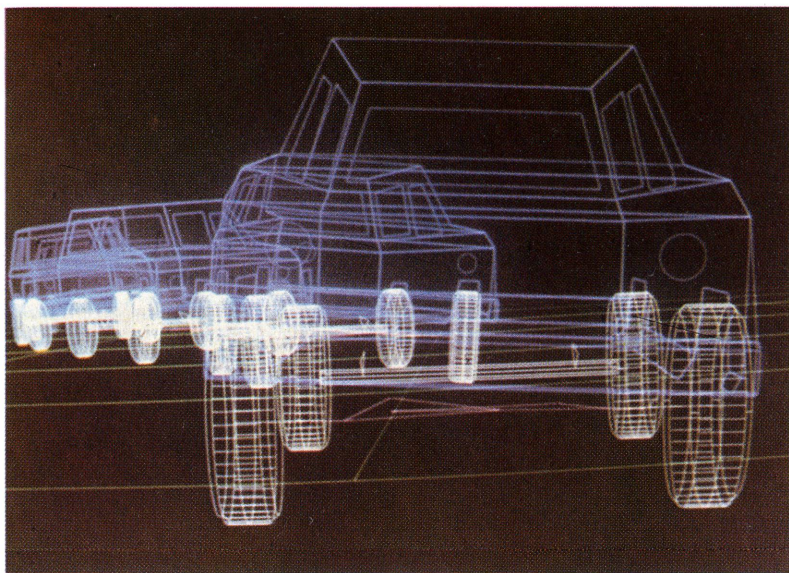
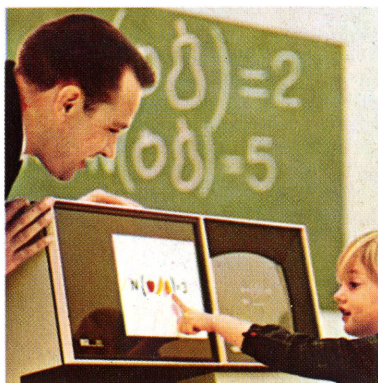
Esistono programmi che permettono a un compositore di registrare una partitura completa e di ascoltarla poi eseguita automaticamente (sotto controllo del calcolatore) da un gruppo di strumenti elettronici, a ciascuno dei quali è assegnata una parte specifica (a fianco Doris Norton, fra micro Apple e sintetizzatori che formano la sua orchestra). Bastano poche istruzioni per modificare i suoni di ciascuno strumento e riascoltare il brano con timbri del tutto diversi. Alcuni sistemi permettono anche di stampare su carta la partitura.

selle permesse, le dimensioni massime delle singole caselle, il tipo e il numero di funzioni matematiche abilitate e, ovviamente, la velocità.
I programmi di tabellone elettronico possono avere le applicazioni piú svariate: sono utilissimi in tutte le situazioni in cui si vorrebbe sapere "che cosa succederebbe se...". Si imposta il tabellone con tutti i suoi parametri, inserendo i valori reali; poi se ne possono creare tutte le varianti desiderate, modificando i valori di singoli parametri e chiedendo al programma di ricalcolare sulla nuova base i totali e i valori delle varie caselle che contengono formule.
Cosí, per fare un esempio, sarebbe abbastanza semplice riuscire a stabilire quale sarebbe lo sconto praticabile a un rivenditore se, anziché 40 pezzi di un certo articolo, ne ordinasse 200 o 300. Con un tabellone elettronico ben impostato ed efficiente, la risposta può essere data in brevissimo tempo: addirittura nel corso di una telefonata!
Si possono usare i tabelloni elettronici anche per forme di simulazione scientifica: non ci sono vincoli sul significato che si può attribuire alle celle e purché i rapporti fra le celle possano essere espressi mediante formule matematiche si può programmare un tabellone elettronico per qualunque funzione. C'è chi ha studiato con questo tipo di sistema il fenomeno della percolazione (esemplificato per esempio dal passaggio dell'acqua nella polvere di caffè in una macchinetta per il caffè) e chi ha simulato modelli atomici del ferromagnetismo basati sulla polarizzazione dello spin dei singoli atomi...

Programmi di comunicazione

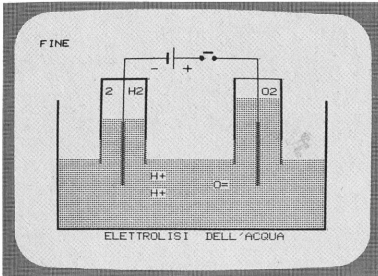
È possibile far "parlare" un computer con altri computer, collegando le macchine direttamente con un cavo, oppure mettendole in comunicazione lungo linee telefoniche specializzate o sulle normali linee telefoniche commutate. È necessario qualche dispositivo fisico opportuno (un'interfaccia per comunicazioni, un modem o un accoppiatore acustico), ma è necessario anche qualche programma che permetta di gestire la comunicazione tra le macchine. E questo perché la comunicazione può avvenire in forme diverse, in funzione delle macchine interessate e del modo di trasmissione: bisogna quindi che ciò

L'informatica ha creato grandi attese anche nel campo dell'istruzione, non solo e non tanto come disciplina autonoma, quanto come strumento per facilitare l'insegnamento e l'apprendimento di altre materie. Le prime esperienze, negli Stati Uniti, nel corso degli anni Sessanta, basate su grandi calcolatori, sono state però troppo ambiziose e deludenti, ispirate a una pedagogia di stampo comportamentista. La disponibilità dei personal computer e una maggiore maturità pedagogica hanno dato migliori risultati, dopo un decennio di stasi, nel corso dei primi anni Ottanta. Costruire software didattico non è comunque un'impresa facile: richiede buona conoscenza dei mezzi disponibili, oltre a una buona conoscenza della materia di cui si vuole parlare, e sotto questo profilo complessivo il materiale soddisfacente disponibile è molto scarso. I risultati migliori, come spesso accade, sono stati raggiunti con spirito pionieristico e un po' di entusiasmo da insegnanti con pochi mezzi, che hanno saputo utilizzare il personal computer per riorganizzare una parte del loro insegnamento, con la partecipazione attiva degli studenti nella costruzione dei programmi. Bastano programmi di poche righe, nelle mani di un buon insegnante, per introdurre argomenti anche complessi di grammatica o di matematica.
I disegni di questa doppia pagina presentano due prototipi di "aula informatica", con i terminali di studenti e insegnante organizzati in rete.

che ciascuna invia in uscita possa essere ricevuto e interpretato correttamente dall'altra. Si tratta, in sostanza, di accordarsi su un "codice" comune e su particolari come la velocità di trasmissione: esistono per questo programmi specifici, che permettono all'utente di predisporre tutti gli elementi per la buona riuscita della comunicazione. Il campo delle comunicazioni a distanza fra computer è ancora relativamente poco sviluppato, ma in continua crescita: si possono realizzare forme efficienti di "posta elettronica", si possono effettuare scambi di materiale a distanza con poca spesa e tempestività. Ancor piú interessante è la possibilità di usare il personal computer per l'interrogazione di grandi banche dati: ne esistono circa 2000 in tutto il mondo, ed è possibile accedervi tranquillamente tramite la rete telefonica, ovunque si trovino.

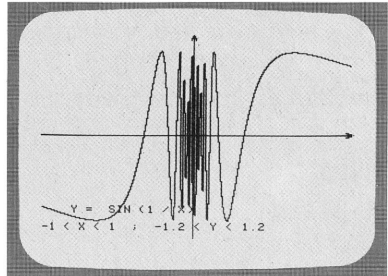
La grafica

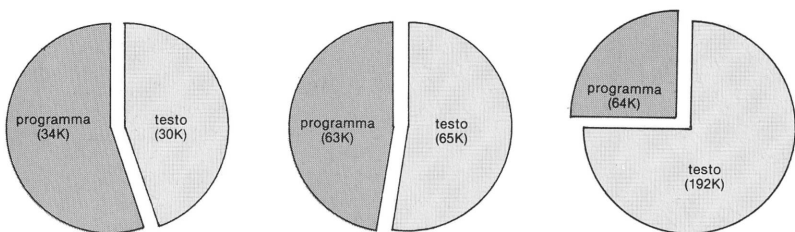
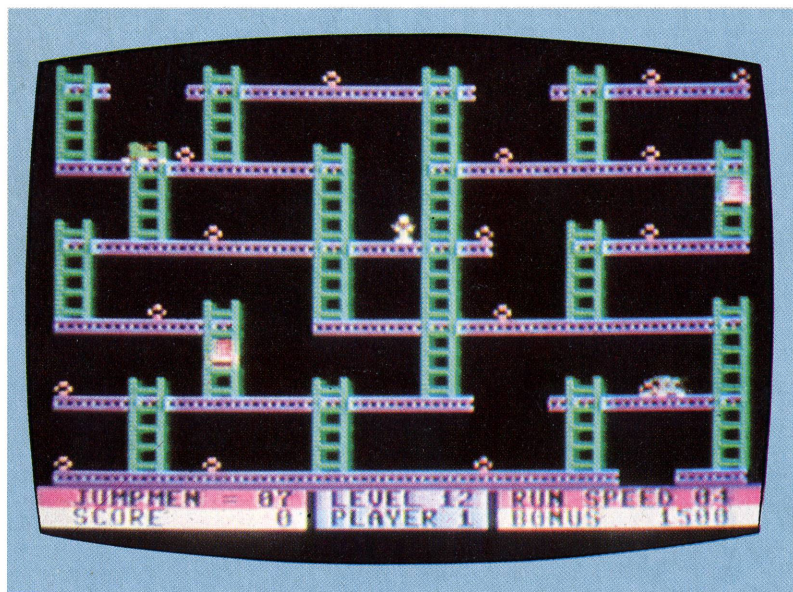
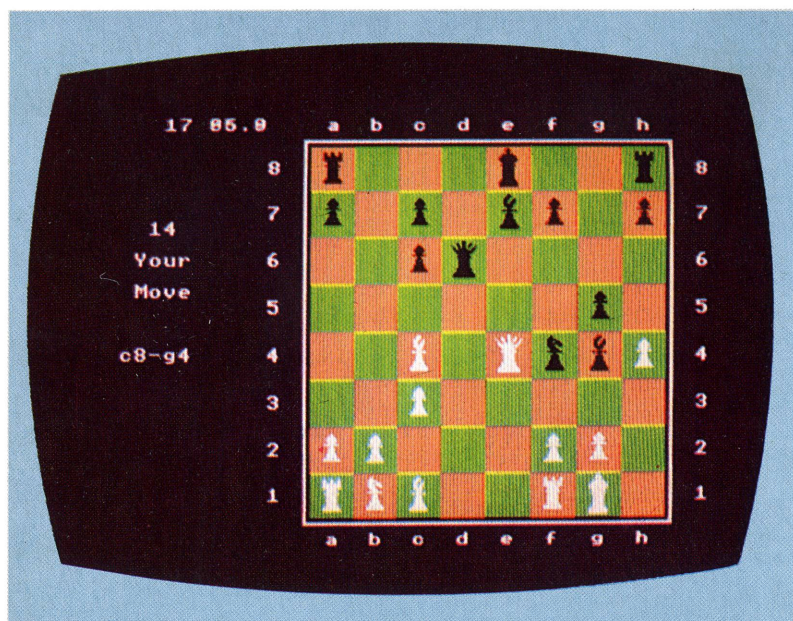
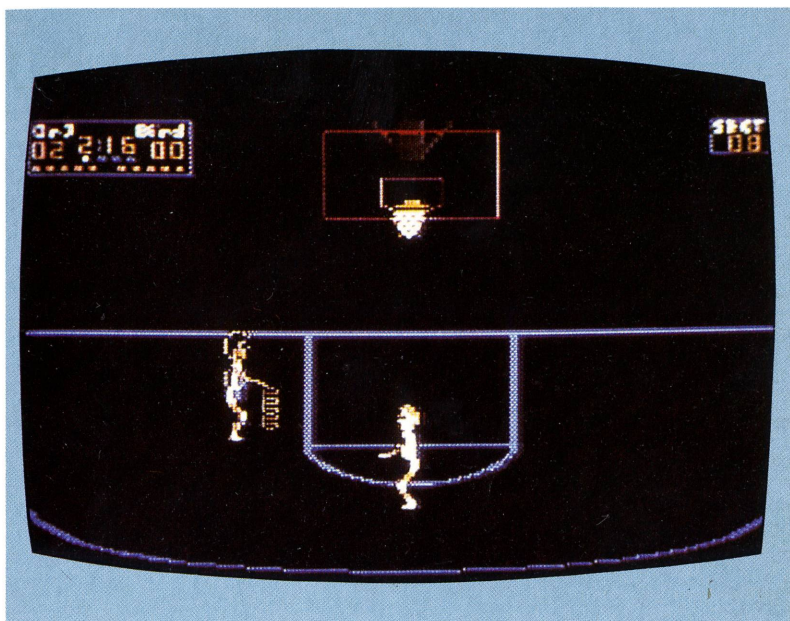
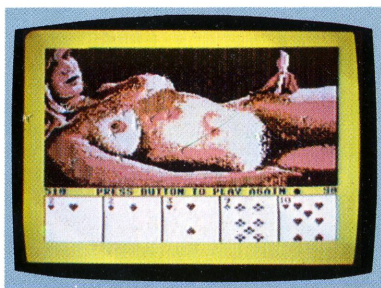
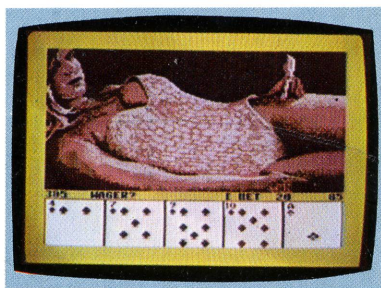
Disegnare con il computer è affascinante: esistono programmi che creano grafici per usi di tipo statistico, altri indirizzati piú specificamente al disegno, con l'uso di strumenti fisici aggiuntivi come tavolette grafiche e digitalizzatori.
L'applicazione grafica piú comune è quella statistico-commerciale: abbiamo dei dati relativamente complessi e vogliamo presentarli in forma grafica, per dare maggiore immediatezza al messaggio. È ormai ben noto che un istogramma o un diagramma a torta può esprimere una situazione con un'efficacia enormemente maggiore, rispetto a una tabella di numeri.
I programmi di questo tipo debbono, normalmente, essere affiancati a programmi di archiviazione o di tabellone elettronico: i dati vengono memorizzati ed elaborati con questi programmi, poi vengono passati al programma di grafica per arrivare a una rappresentazione visiva. Spesso il programma di grafica fa già parte integrante di un programma di archiviazione o di tabellone elettronico, oppure è previsto come "modulo aggiuntivo" a uno di questi programmi. Bisogna sempre tener conto del problema

Fra le attività che il computer permette di realizzare, e che offrono grandi possibilità nella progettazione, nella ricerca scientifica e nell'educazione, vi è soprattutto la simulazione, che permette di eseguire velocemente esperimenti e prove con modelli visualizzati sullo schermo. Nella pagina a fronte si vede una simulazione di un processo fisico (l'enucleazione favorita da impurità sulla superficie di un cristallo); qui sopra, la simulazione di un furgone che permette di evidenziare la deformazione delle sospensioni quando il veicolo cambia direzione. Il pregio della simulazione è la possibilità di eseguire esperimenti difficili o pericolosi a basso costo e con sicurezza: la bontà della simulazione però sta tutta nella validità dei programmi.
A destra, un alunno con il suo computer.

della compatibilità: non è detto che qualunque programma di grafica possa accettare i dati da qualunque altro programma di elaborazione.
Al di là dei diagrammi e di queste forme elementari di *business graphics*, utili soprattutto agli statistici e ai manager d'azienda, esiste una gamma amplissima di programmi per la creazione di immagini, di carattere generico o specializzati. Esistono programmi per la progettazione: sono diffusi soprattutto a livello di grandi calcolatori, ma se ne trovano ormai anche versioni per personal computer, almeno per progetti di piccole dimensioni.
Esistono programmi di grafica per architetti di interni: si possono visualizzare per esempio piante di appartamenti e sagome di mobili, in modo da progettare la disposizione di un interno. Anche in questo caso l'utilità del computer sta soprattutto nella facilità e nella velocità con cui consente di rispondere a domande del tipo "cosa succederebbe se...": che cosa succederebbe mettendo quel mobile contro quella parete e spostando il pianoforte nell'altra stanza? O cose simili.
Tavolette grafiche anche di costo molto contenuto, con i relativi programmi, permettono di disegnare sulla tavoletta come su un pezzo di carta: il disegno viene trasferito sul video, può essere memorizzato ed elaborato in vario modo. La tavoletta svolge sostanzialmente la funzione di "digitalizzatore": trasforma il disegno in numeri binari. Invia all'unità centrale del calcolatore le informazioni relative alla posizione dei singoli segni in termini di coordinate; una volta che il disegno è tradotto in informazioni numeriche (cioè "digitali") è possibile elaborarlo come si elabora qualsiasi informazione numerica. I programmi che accompagnano la tavoletta permettono di sottoporre a trasformazioni geometriche, di assegnare e modificare colori...

Musica

Qualche macchina è dotata all'interno di circuiti di generazione sonora, in ge-

nere relativamente semplici; praticamente in tutte le macchine possono essere inserite delle schede specializzate per la generazione di suoni, anche molto raffinate. In questo modo il computer può essere trasformato in un sintetizzatore musicale digitale (pilotabile eventualmente mediante una tastiera musicale esterna). I programmi di tipo musicale specializzano il computer: si va da semplici programmi per l'insegnamento della musica, magari sotto forma di giochi per i bambini piú piccoli, a programmi che permettono di esercitarsi nel riconoscimento (una sorta di "dettato musicale" automatizzato); da programmi per la generazione di timbri a programmi di ausilio per la composizione; da programmi che trasformano il calcolatore in un sequencer (un dispositivo per memorizzare sequenze di note e rieseguirle a piacere) a programmi orientati alla scrittura e alla stampa di musica. Ce n'è anche qui per tutti i gusti e per tutte le borse.
Cose davvero interessanti (musicalmente) si possono ottenere disponendo di un personal computer dotato di un'interfaccia standard MIDI (Musical Instruments Digital Interface, ovvero interfaccia digitale per strumenti musicali) e di sintetizzatori o altre apparecchiature musicali dotate dello stesso tipo di interfaccia (vedi la didascalia a pagina 53). Esistono programmi che permettono allora di trasformare il computer in un "centro di controllo" che può memorizzare interi brani e rieseguirli automaticamente, assegnando le varie parti ai vari strumenti disponibili e "dirigendoli" nell'esecuzione.

Queste tre immagini sono tratte da altrettanti programmi didattici di fisica e di matematica per la scuola media superiore, realizzati per l'Apple II. La grafica svolge sempre un ruolo importante nel software didattico, specialmente nei casi in cui si simulano esperienze (come, qui sopra, per l'elettrolisi dell'acqua).

La quantità di memoria interna di un computer costituisce un vincolo per le dimensioni e l'efficienza dei programmi che si possono mandare in esecuzione, nonché sulla quantità di dati che possono essere elaborati come un tutto unico. I diagrammi qui sotto si riferiscono al caso di un programma di word processing, ma considerazioni analoghe valgono per qualunque altro programma. Nel caso di una macchina con 256 K di memoria centrale, spesso la quantità di testo elaborabile in un tutto unico rimane di 64 K circa: molti programmi lasciano libera una metà della memoria di lavoro per motivi di efficienza, per avere disponibile per operazioni di trasferimento o per l'immagazzinamento di risultati intermedi o transitori. I programmi integrati più complessi possono avere un'occupazione di memoria anche di 256 K, richiedendo così per un buon funzionamento un minimo di 512 K di memoria centrale.

La didattica

Non c'è dubbio che il computer potrebbe svolgere una funzione di grandissima utilità nel campo della didattica, e non solo per discipline affini all'informatica o nell'ambito delle discipline scientifiche, ma anche per le materie umanistiche (la distinzione è stata finora molto criticata, ma, in particolare nella scuola italiana, rimane di fatto bene viva). Potrebbe essere usato come ausilio per la preparazione di lezioni, come un audiovisivo potente e versatile: esistono linguaggi di programmazione, chiamati *linguaggi autore*, che consentono a un insegnante di realizzare lezioni di qualunque materia sul calcolatore, in modo semplice, molto vicino alla scrittura normale. Si possono realizzare con questi strumenti anche test a scelte multiple, esercizi di ripasso o di valutazione. Molto sta all'abilità dell'insegnante, ma si tratta ancora di strumenti relativamente poveri, che usati superficialmente possono risultare più noiosi e meccanici dei vecchi libri di testo.
Osservazioni analoghe si possono fare per molti programmi confezionati per l'apprendimento di argomenti particola-

I giochi per i personal computer sono spesso adattamenti dei piú famosi giochi elettronici da bar, ai quali ormai hanno ben poco da invidiare: anche un calcolatore domestico di basso prezzo dispone quasi sempre dell'hardware per la gestione del colore, e non sono necessarie grandi capacità di memoria per programmi di gioco anche complessi. Per quel che riguarda i contenuti, la scelta è amplissima: ai giochi spaziali si sono affiancati giochi di avventura di ogni tipo, giochi che simulano sport (dal tennis, fra i primi a raggiungere gli schermi dei personal computer, al calcio, al decathlon, alla pallacanestro: nell'esempio qui sopra riportato, i programmatori hanno dotato gli atleti di "effetto fatica" ed "effetto recupero"), per arrivare fino ai giochi "a luce rossa" (con un po' di fantasia) come lo Strip Poker che si vede qui a fianco. Un settore particolare è quello dei giochi tradizionali, spesso fatti oggetto di ricerca serissima da parte degli studiosi di intelligenza artificiale: dama, poker, scacchi, backgammon. L'interesse degli studiosi per questi giochi sta nel fatto che permettono di cimentare la macchina in attività che richiedono intelligenza e capacità di decisione, ma sono ben definite e governate da regole precise.

TECNOLOGIA

generazione	elemento caratteristico	*paramenti tipici* velocità	dimensioni	dissipazione	affidabilità	costo
prima	valvola	1	1	1	1	1
seconda	transistore	10	1/100	1/1000	100	1/10
terza	circuito integrato	100	1/100	< 1/1000	100	1/100

ELABORATORE

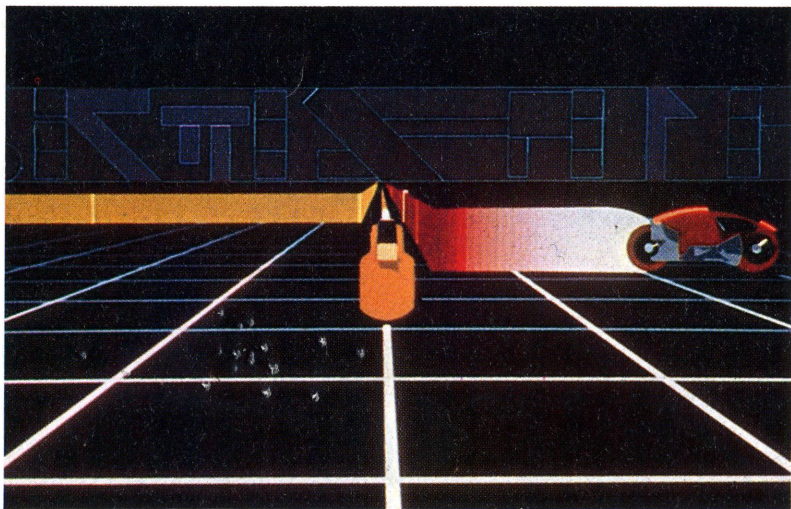
generazione	*unità centrale* governo	memoria	*ingresso/uscita* connessioni	supporto caratteristico
prima	logica cablata	a una porta	dirette	scheda perforata
seconda	matrice logiça di sequenza	a piú porte	con polmone	nastro magnetlco
terza	microprogrammato	a piú livelli	unità di canale	disco magnetico

generazione	modo operativo	linguaggi di programmazione	gestione del sistema	diagnostica
prima	locale seriale	macchina o simile	operatore	manuale
seconda	locale parzialmente simultaneo	simbolici	operatore + software	aiutata dalla macchina
terza	locale e a distanza simultaneo (multi progr.)	alto livello	software (slstemi operativi)	sostanzlalmente automatlca
	tempo reale partiz. di tempo	conversazionali		

ri: esiste molto materiale in lingua inglese, quindi non utilizzabile effettivamente nell'insegnamento scolastico. Anche di questo materiale molto è comunque di qualità discutibile. L'impressione che si prova davanti a questi programmi è che non ci siano ancora idee molto chiare sul modo migliore per sfruttare le grandi capacità del calcolatore, e che invece si tenti di trasferire alla nuova tecnologia modi di lavorare piú adatti per la carta stampata, con risultati ovviamente mediocri.
Laddove la potenza del personal computer si comincia a vedere in azione è nei programmi cosiddetti di "simulazione". Simulare significa riprodurre le caratteristiche essenziali di una situazione, di una attività o di un oggetto attraverso un modello: l'esempio tradizionale piú tipico è costituito dai modelli in scala ridotta di automobili o di aeroplani che vengono sottoposti a test nelle gallerie del vento per valutarne le caratteristiche aerodinamiche. Il modello che viene sottoposto a test è analogo al veicolo reale sotto tutti i punti di vista che interessano: proporzioni, peso relativo, caratteristiche dei materiali costruttivi. Tutto ciò che è inessenziale ne è invece assente: è del tutto ininfluente la forma dei sedili o la disposizione della strumentazione sul cruscotto.
Nel caso della simulazione su calcolatore, il modello è astratto: è dato in realtà da un insieme di equazioni o di relazioni matematiche incorporate in un programma, con una visualizzazione sullo schermo piú o meno realistica. Da tastiera si possono modificare valori e parametri e il programma simula il risultato delle modifiche introdotte, calcolando la situazione risultante. Un esempio tipico, assai diffuso sotto forma di gioco, è il simulatore di volo: sullo schermo si vede la plancia di controllo di un aeroplano, con tutta la strumentazione: con opportuni comandi si può "pilotare" come in una cabina reale. Gli strumenti indicano la velocità, il numero di giri dei motori, l'altezza a cui ci si trova, la posizione degli alettoni e via dicendo: sopra il quadro di comando si può avere anche uno scorcio del paesaggio esterno.
Il programma permette di sperimentare il volo senza rischi: ci si può alzare in volo, girare a piacere e provare ad atterrare. Se ci si schianta per una manovra errata, a differenza di quel che accade nella vita reale, si può riprovare: è questo uno dei pregi essenziali della simulazione. Benché sia visto normalmente come gioco, il simulatore di volo è una versione limitata di un programma completo utilizzato effettivamente per l'istruzione dei piloti: è dunque un esempio di programma d'istruzione.
È relativamente facile trovare programmi per la simulazione di situazioni piú semplici, come ausilio per la didattica di discipline come la fisica o la chimica: si simulano esperimenti di laboratorio, in sostanza, con il vantaggio di poter fare sperimentare agli studenti anche situazioni che in laboratorio andrebbero accuratamente evitate. Se l'attrezzatura simulata salta in aria, nessuno si fa male!
Sotto un profilo del tutto diverso, è estremamente interessante quanto si può fare in campo didattico con il personal computer e un linguaggio di programmazione come Logo; ne parleremo meglio nel capitolo 5, a pagina 65.

La breve storia del calcolatore elettronico viene spesso suddivisa in quattro "generazioni" di macchine, profondamente diverse fra loro. La prima generazione risale agli anni Cinquanta, ed è caratterizzata dalla tecnologia delle valvole termoioniche e dalla scheda perforata come supporto delle informazioni. La seconda generazione, che inizia sul finire degli anni Cinquanta, è caratterizzata dalla tecnologia del transistor, dal nastro magnetico come supporto e dalla comparsa di numerosi linguaggi di programmazione di alto livello. I circuiti integrati e il disco magnetico caratterizzano la terza generazione, iniziata con la seconda metà degli anni Settanta. L'integrazione su larga scala e il raffinamento del software sono invece le caratteristiche della quarta generazione, quella attuale, che data dalla seconda metà degli anni Settanta.
Già si comincia a parlare di calcolatori della quinta generazione, i cui prototipi saranno ultimati entro gli anni Novanta (vedi a pagina 168).

Di tutto un po'

La rassegna dei tipi di programmi disponibili per i personal computer non finisce certo qui: abbiamo solo sorvolato alcuni settori particolarmente sviluppati e interessanti. Le cose che non abbiamo visto sono assai piú numerose di quelle che abbiamo potuto scorgere.
Ci sono, innanzitutto, i giochi. È quasi superfluo parlarne: sono fra i programmi meglio pubblicizzati e assomigliano cosí tanto ai videogiochi e ai giochi elettronici che si possono incontrare nelle sale specializzate, che non è difficile avere un'idea di come possano essere. In effetti, le consolle di videogiochi non sono altro che calcolatori molto specializzati: i componenti fisici sono simili, ma semplicemente organizzati in modo da poter esclusivamente seguire quel tipo di programma. E molti giochi da sala di successo hanno avuto versioni (o copie simili) per i calcolatori personali.
Ci sono giochi d'abilità, giochi di destrezza e di competizione, giochi d'avventura e giochi tradizionali come la dama e gli scacchi. Quasi sempre, con i giochi, è consigliabile l'uso del joystick come strumento di controllo (vedi a pagina 95).
Ci sono, poi, tutti i programmi piú specializzati: per particolari categorie di professionisti, per usi scientifici e tecnici. Si spazia dai programmi per la fatturazione e la gestione del magazzino a pacchetti di programmi per la risoluzione di particolari problemi matematici o per il calcolo di strutture in cemento armato, con tutto quello che può stare in mezzo. Per applicazioni di questo tipo si possono trovare annunci su riviste specializzate, ma piú spesso le informazioni piú valide si ottengono attraverso associazioni di categoria, fiere e mostre, convegni, o nel colloquio con i colleghi. Come consiglio generale, è meglio non fidarsi molto di un rivenditore generico per l'acquisto di programmi relativamente specializzati: non c'è nulla di meglio di una prova in prima persona, affiancata dal giudizio di qualcuno che abbia effettivamente sperimentato già il programma nel suo lavoro quotidiano. Il personal computer è uno strumento potente e affascinante ma, proprio perché i suoi utenti d'elezione sono persone senza conoscenze specifiche di informatica, è stato (ed è) oggetto di speculazioni senza troppi scrupoli. È facile in questo campo spacciare per oro colato un oggetto di nessun valore.

Il calcolatore nel cinema: a fronte, un fotogramma dal film Tron, realizzato dalla Walt Disney nel 1982. Qui sopra, il computer alle Olimpiadi di Los Angeles del 1984: il video permette di seguire piú "da vicino" le prestazioni degli atleti.

Ci sono poi i programmi liberi, quelli che si possono trovare su libri o riviste: se ne trovano di tutti i tipi, semplici, complicati, utili, di svago. Il problema è il tempo che si impiega per ricopiarli, quando sono relativamente lunghi: copiare un programma è – bisogna ammetterlo – piuttosto noioso. È facile commettere errori banali di battitura: un punto al posto di una virgola, una lettera al posto di un'altra. E questo è sufficiente perché poi il programma non giri o giri male. Bisogna rileggere tutto con santa pazienza e andare a individuare gli errori: è davvero un gioco di pazienza. Vale comunque la pena di provarci almeno una volta, per capire che cosa significhi ricopiare un programma. Per chi ha tempo e pochi soldi, questo è comunque il metodo piú economico per crearsi una biblioteca di programmi.

I programmi liberi su libri e riviste sono noiosi da ricopiare ma sono utilissimi sotto molti altri punti di vista: un linguaggio come il BASIC (nel quale sono scritti il 99 per cento dei programmi liberi) si può imparare abbastanza rapidamente, e non c'è niente di meglio che studiare i programmi degli altri per imparare a programmare. Basta un'infarinatura degli elementi fondamentali del linguaggio, per riuscire a capire buona parte dei programmi esistenti; e poi è come con le lingue, l'importante è leggere, leggere, leggere (e possibilmente parlare).

Il software integrato

Difficilmente, quando si svolge un'attività professionale, si riesce a segmen-

tare con molta precisione i propri strumenti di lavoro. Non è che in un certo momento si abbia bisogno esclusivamente di scrivere una lettera, dopo un po' esclusivamente di consultare l'archivio clienti e dopo un po' ancora di fare esclusivamente una proiezione del tipo "che cosa succederebbe se...", e via di questo passo. Nella realtà succede sempre il contrario. Proprio nel bel mezzo di una lettera mi viene in mente che dovrei citare quell'informazione che si trova immersa nell'archivio clienti, oppure che in un appunto volante preso qualche giorno fa avevo segnato giusto quella notizia, e subito dopo mi rendo conto che sarebbe meglio aggiungere alla lettera una nota con qualche dato piú preciso, e che mi farebbe comodo fare quattro conti, per non dire che aggiungere un grafico servirebbe a far capire meglio l'essenziale di quel mio tal discorso...

Nella realtà del lavoro, ma anche in quella della vita quotidiana di casa, difficilmente le singole attività sono ben separate le une dalle altre: in genere si mescolano, si intersecano, si rafforzano a vicenda e continuamente. E le nostre scrivanie, anche quelle dei piú ordinati e metodici, lo testimoniano: rispondiamo al telefono mentre sfogliamo un libro, teniamo a portata di mano un blocchetto per gli appunti, l'agenda magari è aperta per ricordarci gli impegni presi, e vicino c'è un orologio per ricordarci che è già tardi. Inevitabilmente, lo stesso tipo di esigenze si trasferisce sul calcolatore. I primi programmi per i personal computer avevano una vita propria, autonoma, ed erano ben distinti gli uni dagli altri: sembrava già di toccare il cielo con un dito, avere a portata di mano strumenti che fino al giorno prima parevano confinati ai centri di calcolo, dominio incontrastato di tecnici in camice bianco.

Poi anche gli utenti di personal computer si sono smaliziati: sono bastati pochi anni perché i pionieri entusiasti e instancabili lasciassero il posto a utenti piú accorti e calcolatori. E anche i produttori di software, scoperta la miniera d'oro, hanno cominciato a farsi concorrenza non solo sui prezzi ma anche sulla qualità e sull'utilità dei prodotti. Il primo progresso è stata la creazione di "linee" omogenee di prodotti: programmi per l'elaborazione di testi, di archiviazione, di tabellone elettronico, di analisi statistica, di grafica, e simili, con una impostazione comune e quindi con due vantaggi essenziali, cioè la possibilità di usare gli stessi dati o di scambiarseli e la omogeneità dei comandi fondamentali. In tutti i programmi di una serie, per esempio, il comando per memorizzare le informazioni è sempre il medesimo: lo si impara la prima volta, con il primo programma, e poi ce lo si ritrova tale e quale nei programmi successivi. È ovvio, non tutti i comandi possono essere comuni; ma è sicuramente comodo che quelli logicamente comuni siano anche identici a tutti gli effetti. Banale, penserà forse qualcuno! E invece si direbbe proprio di no, che sembri banale a noi oggi dal di fuori: si possono trovare ancora in commercio programmi che, pur essendo prodotti della stessa ditta, non rispettano questa regola elementare di coerenza, che si traduce in facilità e semplicità d'uso.

Il passo successivo è stata la vera e propria integrazione: la fusione, cioè, di piú programmi in un unico "pacchetto". Questa evoluzione non è stata semplicemente un'evoluzione del software: è stata resa possibile dall'evoluzione parallela dell'hardware, cioè della struttura fisica delle macchine. Nel giro di pochi anni, a parità di prezzo la potenza dei personal computer disponibili in commercio è piú che raddoppiata. Alla fine degli anni settanta un personal con una memoria di 32K poteva sembrare già eccezionale; alla metà degli anni ottanta i 128K sembravano diventati un requisito minimo. (Non preoccupatevi del significato di questi valori: il K è solo una misura di capacità della memoria interna del calcolatore, e 4K sono il doppio di 2K, come ci si potrebbe aspettare secondo le regole della matematica elementare. Un po' come la cilindrata delle automobili: una 2000 ha cilindrata doppia di una 1000. La cilindrata è indicativa delle prestazioni possibili, ma non in

Il meccanismo delle finestre e dei menú accessibili con il mouse, sul Macintosh.

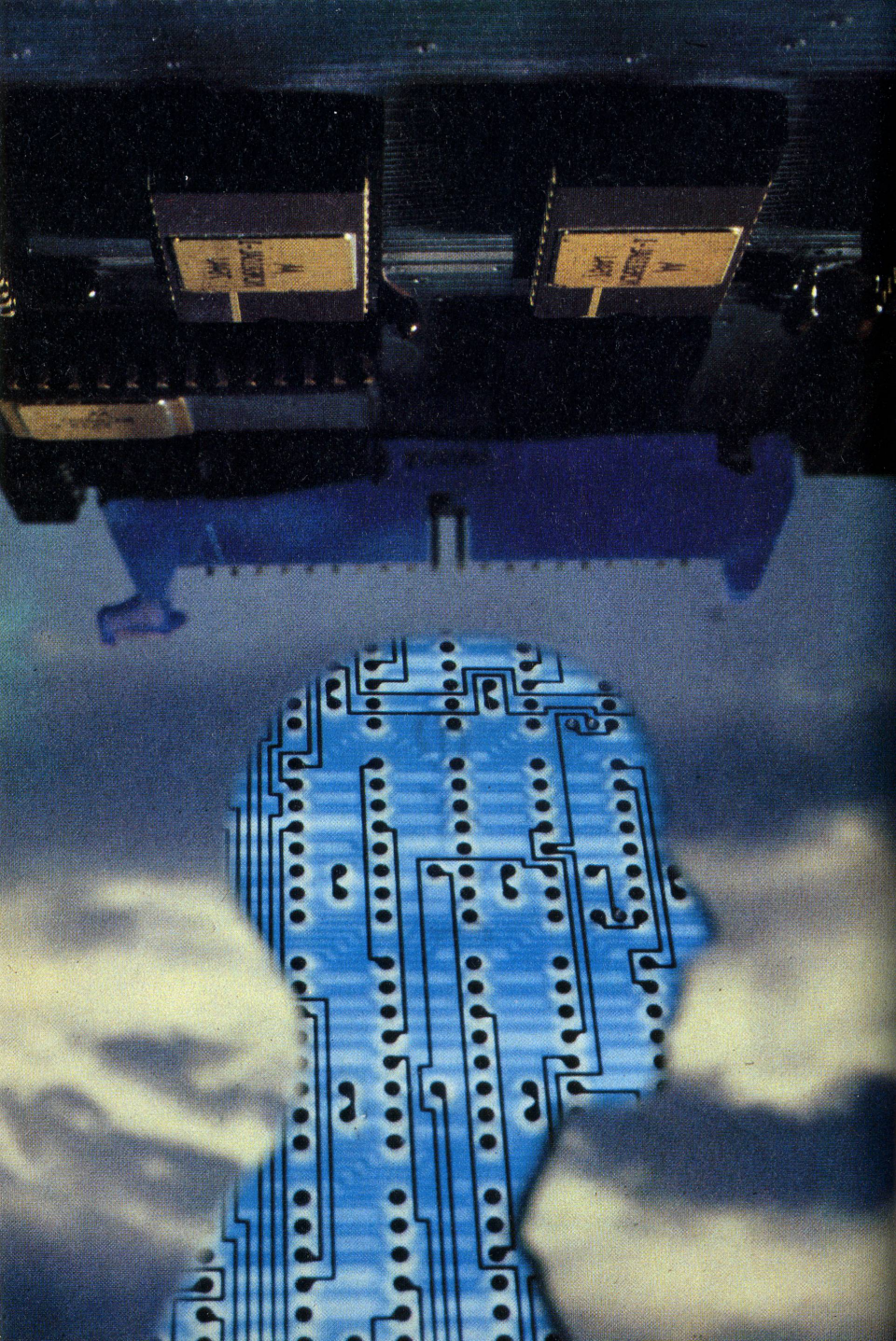
assoluto: non è detto che una 2000 abbia prestazioni doppie di una 1000.)

Un programma può essere eseguito solo se è "caricato" all'interno dell'unità centrale di elaborazione: per la precisione, nella memoria interna dell'unità stessa. E quanto piú è complicato un programma, tanto maggiore è lo spazio di memoria che occupa. Un programma può risultare anche troppo complicato per la memoria interna di una macchina: significa che potrà girare solo su macchine piú potenti. È questo il caso del *software integrato* di piú recente produzione: offre prestazioni eccellenti, ma richiede macchine potenti e con memorie interne di grande capienza.

Nei pacchetti integrati si trovano solitamente le funzioni essenziali che abbiamo visto all'inizio del capitolo: elaborazione di testi, data base, foglio elettronico, grafica, comunicazioni. Il meccanismo grazie al quale questi vari programmi sono integrati è variabile, ma esternamente prende di solito la forma di "finestre" sul video: riquadri che possono essere aperti, chiusi, ingranditi, rimpicciolíti, e che consentono di avere a disposizione contemporaneamente materiali diversi. Per trasferire materiale da una funzione all'altra lo si porta, con comandi opportuni, in una finestra appositamente creata, poi si muove la finestra dove si vuole.

Cosí, la parte principale dello schermo può essere costituita da una finestra su un testo: il capitolo che sto scrivendo, per esempio. Di fianco, molto piccole, posso avere una finestra su un archivio bibliografico in cui ho memorizzato i titoli di articoli pertinenti al tema di queste pagine, il software. In un'altra finestra posso tenere una sorta di blocco di appunti: note sparse prese negli ultimi giorni, intanto che scrivevo queste pagine (note come "devo ricordarmi di parlare dell'argomento tal dei tali, quando arrivo a parlare di questo"). In un'altra finestra ancora, potrei avere un tabellone elettronico. Quando mi serve, porto il mio cursore su una finestra e vado a recuperare le informazioni di cui ho bisogno, senza interrompere il lavoro di scrittura. E, se mi serve, posso "importare" nel testo materiale preso da altre finestre, senza ribatterlo, ma "copiandolo" direttamente, per via elettronica.

Il meccanismo delle finestre si va diffondendo un po' su tutte le macchine: è comodo, molto intuitivo, facile da usare. Esistono alcune macchine, però, tutte basate su questo meccanismo: il Lisa e il Macintosh della Apple, a cui va il merito storico di aver introdotto realmente in prodotti commerciali questo modo di lavorare (le cui origini risalgono piú indietro, in particolare alle ricerche svolte dai laboratori della Xerox a Palo Alto, che tuttavia non hanno mai avuto reali sbocchi commerciali).

L'integrazione può essere ottenuta anche per un'altra strada, almeno entro certi limiti: con programmi particolari che creino una sorta di "ambiente" in cui programmi diversi riescono a interagire, in qualche misura. Questi programmi "integratori" sono abbastanza complessi, e non sempre riescono a raggiungere un grado di efficienza pari a quello di pacchetti integrati costituiti da programmi pensati sin dall'origine per incastrarsi l'uno nell'altro a formare un'unica entità armonica. Possono tuttavia dimostrarsi estremamente utili per chi possieda programmi diversi e voglia usarli insieme, o per chi abbia esigenze specifiche e abbia bisogno di funzioni che solo programmi ben determinati possono dargli, ma al contempo abbia bisogno di collegarli fra loro.

Come si fa a farlo: l'hardware

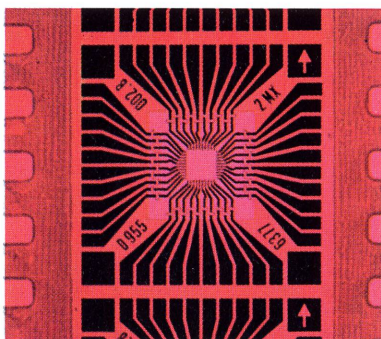
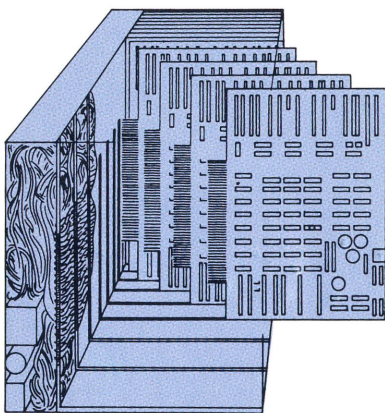
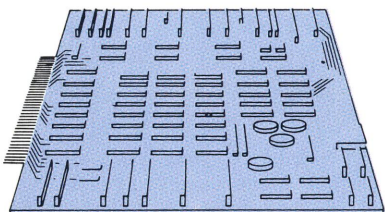
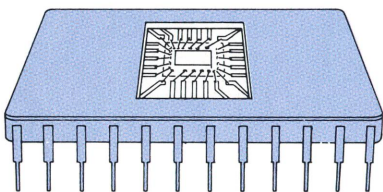
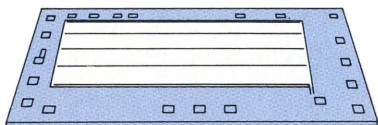
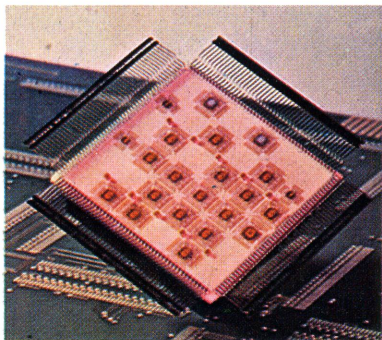
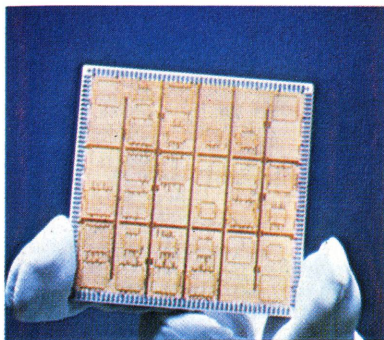
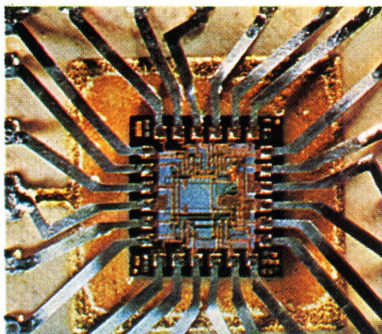
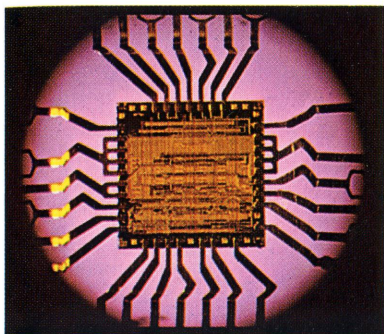
Finora abbiamo dato uno sguardo molto superficiale a "come è fatto" un personal computer e ci siamo invece soffermati piú a lungo sul software disponibile per queste macchine, cioè sul tipo di attività per cui è possibile utilizzarle. È venuto infine il momento di scendere un po' piú nei particolari sulla loro struttura fisica. Cominceremo a vederla dal di fuori: quali sono i componenti macroscopici di un sistema di elaborazione personale, quali le loro caratteristiche distintive, quale è il loro ruolo nell'economia complessiva del sistema.

L'unità centrale

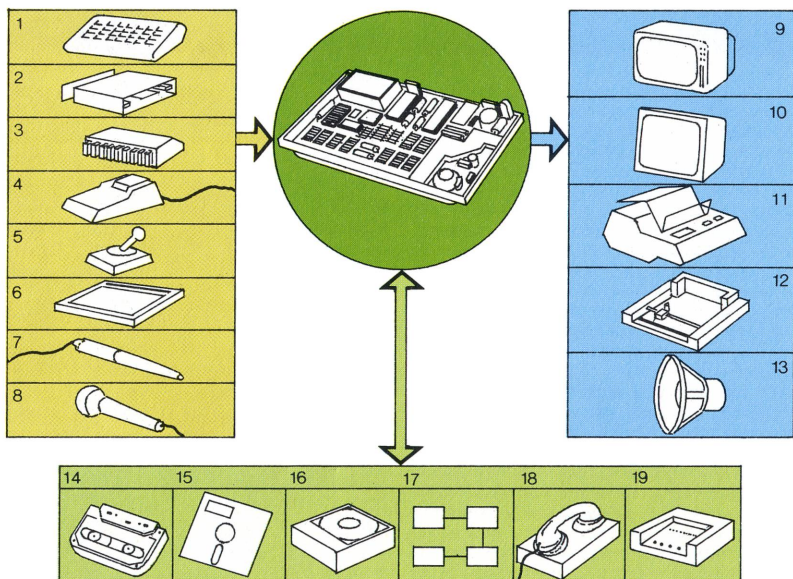
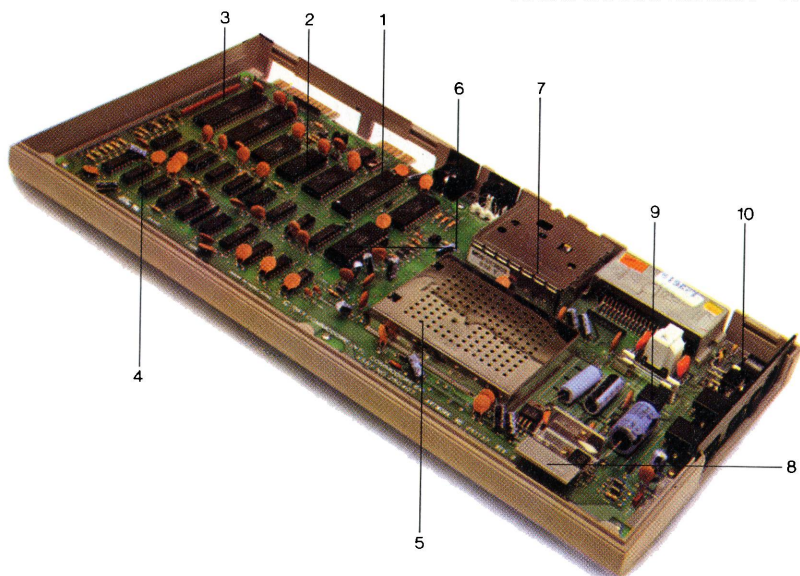
L'unità centrale di elaborazione è il nucleo essenziale del sistema: l'elemento di cui non si può mai fare a meno, per il quale non esistono alternative. L'unità centrale è la sezione in cui vengono svolte tutte le operazioni di elaborazione e ospita, sostanzialmente, la parte "intelligente" del sistema.
Gli elementi che la costituiscono sono un microprocessore, circuiti di memoria, interfacce a cui è demandato il compito di sovrintendere ai rapporti con dispositivi esterni (la tastiera, il video, i drive per dischi, e via dicendo), e tutti i collegamenti fra le varie parti (realizzati con cavi o con conduttori stampati sulle piastre circuitali). Pressoché tutti i componenti sono sistemati su un'unica piastra "madre", sulla quale possono essere predisposti connettori per eventuali espansioni: fessure già munite di una contattiera in cui si possono inserire "schede" a circuiti stampati con ulteriori componenti – componenti non indispensabili al funzionamento del sistema, ma in grado di potenziarlo o comunque di ampliarne le capacità.

Alla base del funzionamento di un computer sta un numero elevatissimo di componenti elementari estremamente semplici.

Il microprocessore è il centro nevralgico: è a suo carico tutto il lavoro significativo. Preleva dalla memoria interna programmi e dati, li elabora, rinvia alla memoria i risultati, gestisce il funzionamento di tutta l'unità. La sua attività è regolata da un orologio o *clock*, che serve a sincronizzare tutti i processi: il clock scandisce il ritmo, rinviando impulsi regolari. Il clock ha una frequenza prestabilita (che si misura in hertz o piú frequentemente in un suo multiplo, il megahertz, pari a un milione di hertz): il singolo ciclo di clock rappresenta il tempo minimo per una operazione elementare del microprocessore (e di conseguenza di tutto il sistema). Questo significa che ogni impulso di clock scandisce l'esecuzione di un'operazione elementare: non possono avvenire piú operazioni elementari all'interno di un unico ciclo. Se dunque il clock ha una frequenza di 4 megahertz (MHz, in sigla), cioè di 4 milioni di cicli al secondo, in un secondo non possono avvenire piú di 4 milioni di operazioni elementari. La frequenza di clock dà pertanto una misura – ancora molto generale – della velocità del microprocessore: un microprocessore a 5 MHz è piú veloce di un microprocessore che funziona a 4 MHz. (L'espressione "microprocessore a 5

MHz" è un'abbreviazione per "microprocessore funzionante con una frequenza di clock di 5 MHz").
Il microprocessore lavora con cifre binarie: una cifra alla volta? No. Tutti gli eventi all'interno dell'unità centrale interessano sempre gruppi di un numero ben definito di bit (cioè di cifre binarie),

Lo stesso elaboratore che non più tardi di 25 anni fa riempiva con i suoi armadi un intero salone, oggi sta tranquillamente su un angolo di scrivania. Il che non significa solo risparmio di spazio, perché il fattore dimensionale condiziona largamente le caratteristiche dell'elaboratore. La velocità operativa, per esempio, dipende dai tempi di propagazione dei segnali elettrici all'interno della macchina e quindi dalle dimensioni geometriche di questa; e una analoga dipendenza si può individuare per la dissipazione di calore, l'affidabilità e via dicendo. La strada principale per la progressiva diminuzione delle dimensioni è stata l'integrazione circuitale, cioè la realizzazione di complessi circuiti elettronici in un chip di silicio. Gran parte dello spazio è tuttavia rubata dalla "confezione" dei chip: tra il volume di un chip nudo e quello del chip montato in un armadio c'è un rapporto da 1 a 10.000.
I chip che costituiscono effettivamente un calcolatore medio-grande non riempiono nemmeno un bicchiere.

Queste sei fotografie presentano in forma di sequenza gli stadi successivi dell'assemblaggio dei chip. In alto a sinistra, uno spezzone della pellicola da 35 millimetri su cui vengono montati i chip: la bobina permette di trasferirli lungo le linee di collaudo e montaggio. In basso a sinistra, uno spezzone della pellicola con la sagoma di rame (ragno) al cui centro verrà collocato il chip. In alto al centro, il chip saldato alle zampe del ragno: in questo modo si distanziano le terminazioni e risulta quindi piú facile lavorare sul chip. In basso al centro: i chip poi vengono montati su un substrato ceramico formato da varie piazzole. Su questa piastrina vengono depositati, mediante un processo serigrafico, vari strati di conduttori e isolanti che formano la rete di interconnessione dei chip. In alto a destra, il chip saldato al substrato ceramico (le zampe del ragno sono ormai bloccate) e, infine, in basso a destra, una piastra ceramica completa dei chip, dalla quale sporgono i connettori mediante i quali potrà essere collegata ad altre.

numero caratteristico del microprocessore. I microprocessori tipici dei calcolatori domestici lavorano con gruppi di 8 bit alla volta (cioè con un *byte* alla volta); i personal piú potenti lavorano con gruppi di 16 o 32 bit alla volta. Il significato delle espressioni "microprocessore a 8 (16, 32) bit" è questo: il microprocessore tratta contemporaneamente ("in parallelo") sempre 8 (16, 32) bit. La quantità di bit che il microprocessore tratta in parallelo definisce la lunghezza di *parola* della macchina. Un personal computer con un microprocessore a 16 bit lavora, cioè, con "parole" di 16 bit. (Può capitare di trovare qualche confusione fra byte e parola: ma byte significa sempre "gruppo di 8 bit" e i due termini diventano sinonimi solamente nel caso dei microprocessori a 8 bit, perché in quel caso la lunghezza di parola è di un byte.)

In un microprocessore a 8 bit, a ogni ciclo di clock possono essere interessati 8 bit; in un microprocessore a 16 bit i bit interessati sono 16. A parità di tempo, dunque, un microprocessore che tratti parole piú grandi può, di massima, svolgere una mole di lavoro maggiore.
Ciò che è ancor piú interessante, tuttavia, è la *capacità di indirizzamento*: il microprocessore deve continuamente accedere alla memoria interna per prelevare informazioni o per depositarvene. Per effettuare un'operazione di "lettura" o "scrittura", però, deve prima comunicare alla memoria in quale cella deve andare a svolgere l'operazione.
Possiamo immaginare la memoria come una grande matrice quadrata di celle, disposte per righe e colonne: ogni cella può essere identificata allora con il numero della riga e della colonna al cui incrocio si trova. Per accedere a una data cella, allora, il microprocessore deve specificare il suo indirizzo: invia prima una serie di bit che rappresentano il numero di riga, e le linee elettriche di quella riga vengono aperte; poi invia una serie di bit che rappresentano il numero di colonna, e vengono attivate le linee elettriche di quella colonna. Solo la cella prescelta ha tutte e due le linee (di riga e di colonna) attivate, ed è cosí l'unica predisposta per la successiva operazione di lettura o scrittura.
A ogni ciclo un microprocessore a 8 bit può inviare solo 8 bit: quindi può effettuare una selezione solo fra il massimo numero di righe (poi di colonne) rappresentabile con 8 cifre binarie. Con 8 cifre binarie si possono rappresentare solo 256 numeri (da 0 a 255), quindi la struttura della matrice di memoria non può essere superiore a 256 righe per 256 colonne. 256 × 256 = 65.536, e 65.536 è il massimo numero di celle di cui può essere costituita la memoria indirizzabile direttamente da un microprocessore a 8 bit.
Per arrivare a capire il tipo di carattere citato comunemente sulle pubblicità e sugli opuscoli illustrativi delle macchine in commercio, dobbiamo fare ancora un piccolo passo, però. Con il sistema appena visto selezioniamo una sola cella alla volta, ma abbiamo appena finito di dire che le parole trattate dal nostro

In questa pagina si può vedere l'interno di uno dei calcolatori domestici piú diffusi, il Commodore 64. I numeri evidenziano il microprocessore (1), le ROM che contengono il BASIC e altro software di base della macchina (2), i chip per il controllo della tastiera e della "porta" per ulteriori collegamenti esterni (3), gli 8 chip di RAM di 8 K byte ciascuno, per un totale di 64 K byte (4), il chip per il controllo della grafica (5), quello per il controllo dei suoni (6), il codificatore per pilotare un normale televisore domestico (7), il regolatore di tensione (8), l'alimentatore (9) e la presa per l'alimentazione (10).
Il diagramma della pagina a fianco mostra vari tipi di dispositivi che possono fungere da unità di ingresso e di uscita per l'unità centrale di elaborazione di un personal.
Tastiera (1), cartucce (2), chip di ROM (3), mouse (4), joystick (5), tavoletta grafica (6), matita luminosa (7), microfono (8): tutti questi dispositivi servono per l'immissione di dati.
Televisore (9), monitor (10), stampante (11), plotter grafico (12) e altoparlante (13) permettono l'uscita dei dati.
Cassette di nastro magnetico (14), dischetti flessibili (15), dischi rigidi (16), reti di calcolatori (17), accoppiatori acustici (18) e modem (19) possono svolgere funzioni sia di unità di uscita, sia di unità di ingresso dei dati.

microprocessore sono di 8 bit e le singole celle di memoria possono ospitare solo uno 0 o un 1, cioè un solo bit. La soluzione è semplice: le celle di memoria possono essere viste disposte su "piani" paralleli: quando selezioniamo una cella con il suo indirizzo, selezioniamo la cella con quell'indirizzo su otto piani paralleli. La parola che viene poi inviata alla memoria, per esempio per esservi "scritta", verrà depositata tutta contemporaneamente, ma con i suoi otto bit su otto piani diversi (ma sempre allo stesso indirizzo in tutti i piani).
Ora dobbiamo fare un po' di conti. 8 bit sono un byte; dunque complessivamente la nostra memoria può ospitare 65.536 × 8 bit, ovvero 65.536 byte. Nel mondo dei computer il byte ha un multiplo, il chilobyte (abbreviato in Kbyte o semplicemente K), cosí come nel mondo dei salumieri il grammo ha come multiplo il chilogrammo. Il mondo dei salumieri però è tutto improntato al sistema decimale e un chilogrammo è uguale a 1000 grammi (1000 è 10 elevato alla terza potenza); nel mondo dei computer, invece il Kbyte non è pari a 1000 byte, bensí a 1024 byte: il mondo

dei computer è improntato al sistema binario, e 1024 è la potenza di 2 piú vicina a 1000 (1024 è pari a 2 elevato alla decima potenza).
Siamo arrivati alla fine: 65.536 è pari a 1024 × 64, quindi 65.536 byte sono 64 Kbyte o, piú sinteticamente, 64K.
Tutto il discorso si riassume dunque in queste poche parole: un microprocessore a 8 bit può indirizzare direttamente al piú 64K di memoria interna.
Seguendo un ragionamento analogo, si può vedere come un microprocessore a 16 bit (o addirittura uno a 32 bit) possano indirizzare un numero nettamente piú elevato di celle di memoria (la proporzionalità non è, in realtà, lineare, perché si presentano altri problemi, tuttavia le differenze rimangono nettissime).
Può capitare di trovare macchine con microprocessore a 8 bit che denunciano capacità di memoria superiori ai 64K: in questo caso il costruttore ha messo in atto un piccolo trucco. Il microprocessore può indirizzare direttamente al massimo 64K di memoria, ma è pensabile di predisporre piú di un "banco" di 64K e un meccanismo per far selezionare al microprocessore a quale banco vuole indirizzarsi. Il meccanismo non è per niente efficiente e non viene utilizzato molto spesso: in effetti, il microprocessore "vede" sempre solo 64K e bisogna specificare espressamente nei singoli programmi quando deve "girare la testa" e "vedere" un altro gruppo di 64K. Alla fine, i vantaggi non sono eccezionali.

I microprocessori esistenti, adatti per i personal computer, non sono moltissimi: alcuni, poi, hanno concentrato l'interesse dei costruttori e sono diventati una sorta di "standard", almeno per alcune fasce di mercato. Cosí è stato, per esempio, per il 6502 della MOS Technology (intorno al quale sono state costruite le varie versioni dell'Apple II e l'Apple III, il Commodore 64 e varie altre macchine della Commodore Business Machines) e, soprattutto, lo Z80 della Zilog (derivato dall'8080 dell'Intel Corporation), per quel che riguarda il mondo degli 8 bit. Lo Z80, in particolare, si è affermato come uno standard grazie anche alla disponibilità di uno specifico sistema operativo, il CP/M (Control Program/Microcomputer) della Digital Research, che ha riscosso i consensi di moltissimi utenti. Il sistema operativo fa parte del cosiddetto "software di base" o "software di sistema": quell'insieme di programmi senza i quali il calcolatore non può fare assolutamente nulla, che gestiscono le sue attività di base e i suoi rapporti con le varie unità esterne. Un sistema operativo coordina tutte le risorse fisiche e logiche del calcolatore: è un po' il fondamento su cui si può edificare ogni altro programma (dal compilatore per un linguaggio di alto livello al programma per tenere la contabilità domestica).
Nel mondo dei 16 bit hanno avuto un successo particolare i microprocessori della Intel 8086 e 8088, internamente sostanzialmente identici, grazie all'adozione dell'8088 da parte della IBM nel suo Personal Computer. L'8088 è un microprocessore che lavora internamente a 16 bit, ma è poi collegato al mondo esterno (alla memoria in particolare) mediante una "pista" per la trasmissione di dati a 8 bit, il che lo rende piú potente di un microprocessore a 8 bit, ma non tanto veloce ed efficiente quanto un microprocessore completamente a 16 bit (cioè funzionante a 16 bit sia internamente, sia nei rapporti con l'esterno). L'8086 ovvia a questa limitazione: è un microprocessore totalmente a 16 bit ed è stato ampiamente utilizzato da tutti i concorrenti della IBM che hanno cercato di realizzare macchine compatibili con il Personal Computer della grande casa americana, ma al contempo con qualche pregio in piú.
Nel mondo dei 32 bit, oggi il microprocessore di maggior successo è il 68000 della Motorola, utilizzato nel Lisa e nel Macintosh della Apple e in varie altre macchine di grande potenza, che si situano, per prezzo e prestazioni, al confine con la fascia dei minicomputer.

La memoria interna. La memoria interna dell'unità centrale è un elemento qualificante: è il "magazzino" deputato

alla conservazione di tutti gli elementi logici necessari per l'attività utile del calcolatore: programmi e dati. Il computer può eseguire un programma solamente quando è nella sua memoria centrale: se si possiede un programma su disco o su cassetta, il primo passo per poterlo eseguire è sempre quello di "caricarlo" nella memoria centrale.
Ne abbiamo già accennato, ma è bene ribadire il concetto. "Memoria" è un termine generale, per indicare tutti i dispositivi che svolgono la funzione di conservare, a beneficio del calcolatore, informazioni e dati. Bisogna poi distinguere tra due tipi di memoria: la memoria principale e la memoria secondaria.
Memoria principale è quella interna alla macchina, e ha forma di circuiti elettronici: è relativamente costosa ma permette operazioni di lettura e scrittura molto veloci.
Memoria secondaria o *memoria di massa* è quella esterna all'unità centrale, basata su supporti in genere di tipo magnetico, cioè nastri o dischi: ha costi inferiori (in termini di costo per unità di informazione memorizzabile), ma le operazioni di lettura e scrittura sono molto piú lente.
La memoria principale è basata sullo stesso tipo di tecnologia dei microprocessori: integrazione a larga scala, ormai anche a larghissima scala, su piastrine di silicio. Al di là della tecnologia usata, si distinguono oggi due forme di memoria principale: quella disponibile all'utente, in cui cioè è possibile scrivere e leggere informazioni, e quella dalla quale è possibile esclusivamente leggere. La prima è la RAM (sigla per *Random Access Memory*, memoria ad accesso casuale), la seconda è la ROM (*Read Only Memory*, memoria a sola lettura). La ROM è protetta nei confronti della scrittura: l'utente può solo andare a prelevare informazioni, non può nemmeno accidentalmente andare a memorizzarne altre. Questo significa che viene fornita "a scatola chiusa" all'origine: le informazioni che conserva vi sono state depositate dal costruttore.
Abbiamo detto molte volte che il computer in sé è tabula rasa, non è in grado di fare nulla fino a che non viene dotato di un programma. Esistono tuttavia una serie di operazioni fondamentali che la macchina deve comunque svolgere, all'atto dell'accensione, come predisporsi ad accettare segnali in ingresso da una tastiera, istituire il collegamento con un'unità di uscita come il video, su cui inviare sempre una "eco" di tutto ciò che succede, e via dicendo. Il programma necessario per effettuare queste operazioni elementari ma indispensabili è memorizzato in una ROM, e viene caricato automaticamente al momento dell'accensione. In gergo questa operazione di avviamento viene definita *bootstrap*, vocabolo che si può tradurre con "lancio", ma in realtà significa "laccio da scarpe": il termine inglese allude a un modo di dire che significa piú o meno "tirarsi su con i lacci delle proprie scarpe" (qualcosa di simile faceva il Barone di Münchhausen nel tentativo di raggiungere la Luna). Vi capiterà sicuramente di sentire qualcuno che parla di "bootstrappare" il sistema, e quel che vuol dire è proprio questo: farlo partire da zero, avviarlo, lanciarlo con i programmi che si autocaricano e lo predispongono a svolgere qualche attività ulteriore. (Vi sconsiglierei di usare a vostra volta quel termine: è proprio molto brutto e gergale!).
A parte i programmi fondamentali di lancio, la ROM di un personal computer può contenere altre cose: quasi sempre contiene anche un interprete BASIC, alcuni circuiti contengono il disegno dei caratteri che vengono visualizzati su video (ROM dei caratteri) e via dicendo. La capacità della ROM si misura in bit come quella di una RAM, piú comunemente in byte o in Kbyte. Quando leggete su un opuscolo che il computer tal dei tali possiede 8K di ROM, vuol dire che il costruttore fornisce, insieme con l'hardware, già inseriti in macchina, 8 × 1024 byte di programmi. Questi programmi si dicono anche "residenti" in macchina. La quantità di ROM di cui una macchina è fornita dà una misura della quantità e complessità dei programmi che vi mette a disposizione direttamente: una macchina con 24K di

*Un registratore a cassette costituisce la forma piú economica di memoria di massa: i suoi svantaggi piú evidenti sono la lentezza e la caratteristica di "sequenzialità"; bisogna far scorrere infatti tutto il nastro per trovare l'ultima informazione che vi è stata registrata. Le cartucce (*cartridge*, in inglese) sono chip di memoria a sola lettura (ROM) su cui sono registrati programmi, montate in piccoli contenitori protettivi muniti di connettori per il collegamento a un computer (foto a destra): costituiscono un mezzo comodo per la commercializzazione di software soprattutto per i piccoli calcolatori che non dispongono di drive per dischi: grazie alla cartuccia il programma può essere caricato velocemente, ed è facile la protezione da copie non autorizzate.*

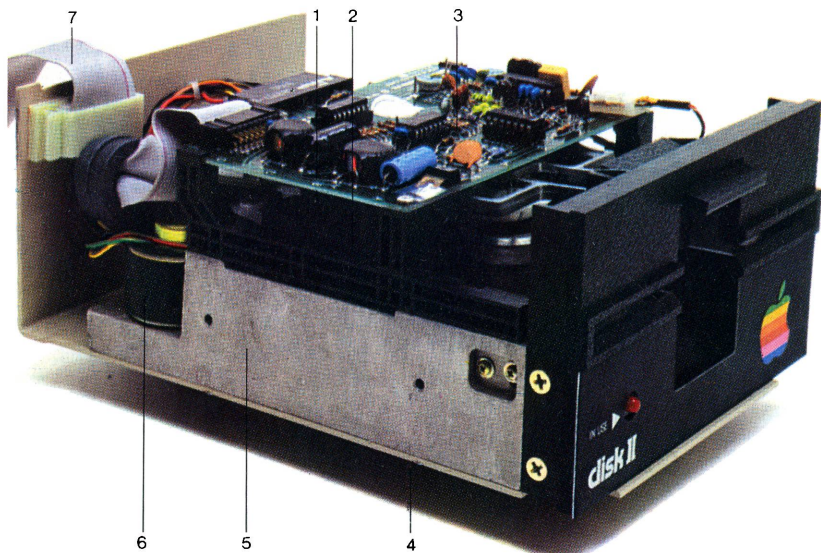
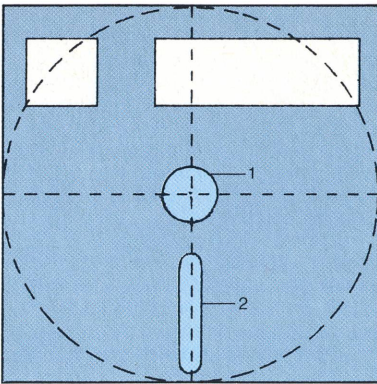
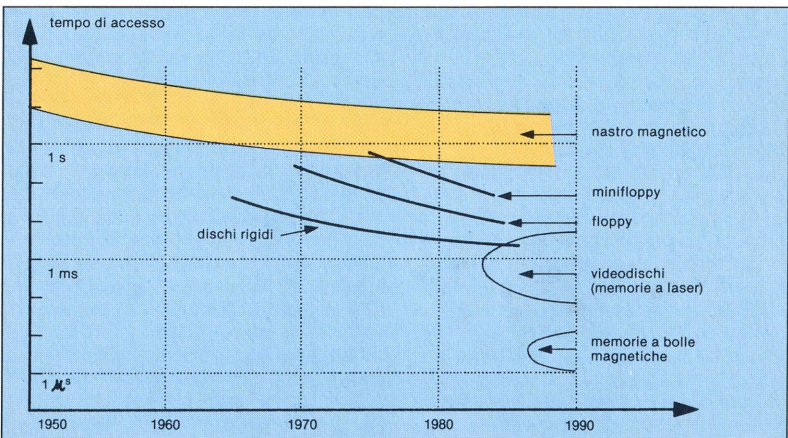
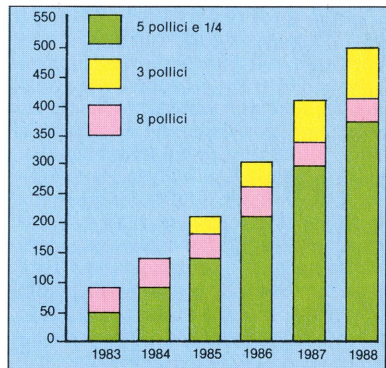
ROM possiede internamente una dotazione di programmi molto piú ricca di un computer che ha solamente 8K di ROM. La dotazione di ROM aumenta molto nel caso delle macchine portatili, quelle che stanno in una ventiquattrore, per intenderci: per rendere il piú possibile autonoma e autosufficiente la macchina il costruttore deve dotarla internamente di tutto il software possibile.

In ROM è possibile memorizzare qualunque programma: l'interprete BASIC come un programma di word processing o di data base. Il costo della ROM, però, è elevato, e non è possibile piú modificare i suoi contenuti: si può solo togliere tutto il chip e sostituirlo con un altro. Questo rende ovviamente piú difficoltoso l'aggiornamento dei programmi, oltre a far aumentare il costo dell'-hardware di base, e di conseguenza ha senso soprattutto per i computer portatili, dove la compattezza deve essere la caratteristica principale.

Sapere che cosa ha in ROM un computer è dunque importante: se è già disponibile un interprete BASIC, significa che non è necessario acquistarlo separatamente su disco. Questo influisce sui prezzi e deve influire correttamente anche su una decisione d'acquisto. La presenza di un interprete BASIC in ROM, comunque, non esclude la possibilità di un acquisto separato di un interprete BASIC diverso su disco: questo rimane sempre possibile. La presenza in ROM significa solo la disponibilità immediata: è sempre possibile togliere il controllo della macchina all'interprete interno e affidarlo a un interprete che viene caricato dall'esterno (o genericamente, a qualsiasi altro programma che venga caricato dall'esterno). I programmi caricati dall'esterno vengono depositati, però, in RAM, memoria di lettura e scrittura. La ROM contiene informazioni in forma permanente: togliendo l'alimentazione le informazioni vi restano memorizzate, e saranno nuovamente disponibili alla riaccensione della macchina. La RAM è, invece, una memoria *volatile*: le informazioni vi restano memorizzate solo fintantoché il computer resta acceso e vengono cancellate quando si toglie l'alimentazione. Se caricate un programma dall'esterno, poi spegnete il computer, quando ridate alimentazione non lo troverete piú; e lo stesso vale per i vostri dati. Esiste qualche eccezione: i computer portatili dispongono di RAM non volatile, perché sono pensati per funzionare indipendentemente da un sistema di memoria di massa esterno. La RAM di questi calcolatori è realizzata in una tecnologia microelettronica particolare (che prende il nome di C-MOS): non è del tutto vero, comunque, che questa RAM sia non volatile. In effetti la tecnologia C-MOS permette di realizzare circuiti di memoria che richiedono poca energia di alimentazione, che può essere fornita, an-

ziché dalla corrente di rete, da una piccola batteria al nichel-cadmio (sul tipo di quelle utilizzate anche nelle macchine fotografiche piú recenti). In queste RAM, dunque, le informazioni rimangono memorizzate anche quando la macchina viene spenta: la batteria al nichel-cadmio le mantiene "rinfrescate" per periodi abbastanza lunghi (una settimana, anche piú), largamente sufficienti in genere perché il possessore del calcolatore portatile possa rientrare alla sua sede abituale e "scaricarle" per sicurezza – ove lo voglia – su un supporto di memoria di massa a lungo termine.
La RAM è un magazzino vuoto a completa disposizione dell'utente: quanto piú grande è il magazzino, tanto maggiore è la quantità di istruzioni e di dati che può trovarvi posto. Le dimensioni della RAM costituiscono cioè un vincolo alla complessità dei programmi che vi possono essere caricati, e alla quantità di informazioni che possono essere messe a disposizione immediata del programma.
Quando si comincia a usare seriamente un personal computer, sembra che la RAM non sia mai sufficiente: si vogliono programmi sempre piú raffinati e complessi, e con l'uso tutti gli archivi di dati si ampliano. È sempre opportuno scegliere sistemi che permettono di ampliare la dotazione iniziale di RAM, anche se questo può incidere sul costo.
Questo è uno dei motivi essenziali per cui è meglio individuare prima il tipo di software di cui si ha bisogno, e poi il tipo di hardware piú adatto: tanto per fare un esempio, pacchetti di software integrato come Symphony della Lotus Corporation o Framework della Ashton Tate, che incorporano elaborazione testi, sistema di archiviazione dati, foglio elettronico, grafica e comunicazioni, richiedono una capacità di memoria minima di 384K, e anche cosí permettono di manipolare una quantità abbastanza piccola di informazione, tanta è la quantità di memoria che viene occupata dai soli programmi! Anche con 640K (massima capacità di memoria a cui può essere espanso un Personal Computer IBM, al momento in cui scriviamo) uno di questi programmi riesce a trattare al massimo 800 record di archivio dati oppure un centinaio di cartelle di testo, ma solo 400 record di archivio dati e una cinquantina di cartelle contemporaneamente: la situazione poi peggiora se nello stesso tempo sono attive anche altre funzioni!

Le memorie di massa: cassette e cartucce. I circuiti dell'unità centrale permettono di trasferire una copia dei contenuti della RAM all'esterno, verso qualche sistema che li registri in forma permanente. Il mezzo piú economico, quello piú adatto per i computer di tipo domestico, ma valido anche per altri sistemi (per esempio, in particolare, per i

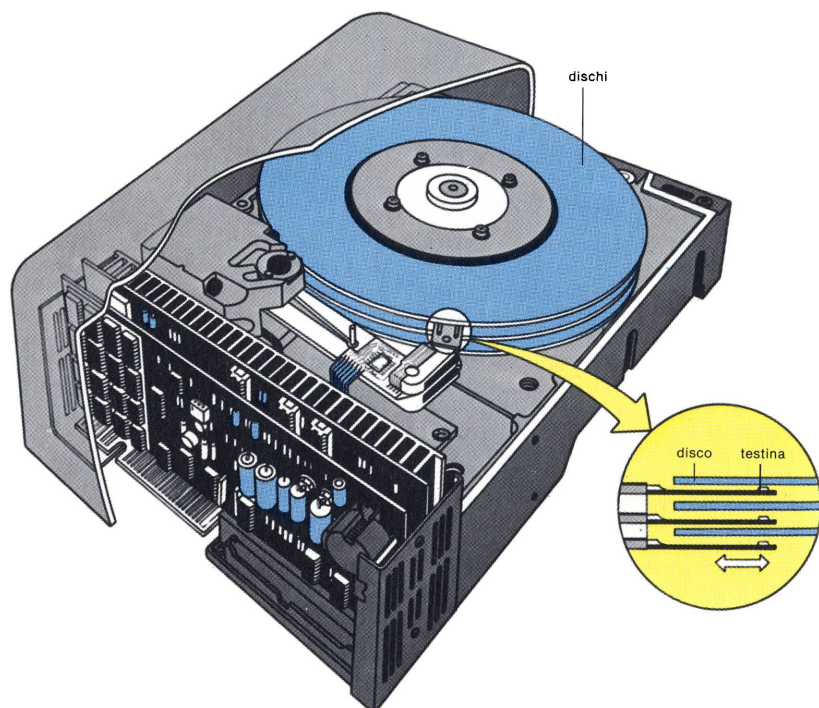
La forma di supporto piú usata per le memorie di massa dei personal computer è il floppy disc, un dischetto di plastica flessibile con la superficie magnetizzata, simile ai dischi audio, ma permanentemente conservato in una busta protettiva. Nel disegno a destra si può vedere come la busta possegga delle piccole aperture che servono rispettivamente come foro di posizionamento iniziale (1) o indice di settore e come punto di accesso (2) della testina di lettura/scrittura al disco stesso. Qui a sinistra si possono vedere due unità (o drive*) per dischetti: i contenitori proteggono la meccanica, che è di grande precisione: si vedono solo la fessura per l'inserimento dei dischi e la griglia per il raffreddamento. Una volta tolte le parti esterne, si possono vedere l'elettronica di controllo (3) e il motore che fa ruotare il dischetto all'interno della sua busta (6). Sotto l'elettronica di controllo sono posizionati il cuscinetto di sostegno del disco (2) e la testina di lettura/scrittura (1) montata su un braccio, collegato al motore passo a passo (5, nascosto dalla piastra); questo motore permette il posizionamento della testina stessa sulla traccia opportuna del disco (2). Sempre dietro la piastra è collocato il volano (4); (7) è il cavo di interfaccia che trasmette i segnali da e verso il computer e porta l'alimentazione per i motori.*

Esistono tre formati di dischetti flessibili: il piú comune per i personal computer è quello da 5 pollici e 1/4; il formato da 8 pollici è piú diffuso sugli elaboratori maggiori. Quelli da 3 pollici (detti anche microfloppy), di piú recente introduzione, stanno progressivamente acquistando popolarità: hanno una custodia rigida e possono avere una capacità anche di 720 K byte (in alto a destra).
Il grafico qui a fianco offre una previsione dello sviluppo del mercato dei dischetti per l'Europa fino al 1988.
Il diagramma in basso dà un'idea di come siano variati, dagli anni Cinquanta, i tempi di accesso per i vari tipi di memoria (videodischi e memorie a bolle sono ancora in fase sperimentale).

calcolatori portatili) è un registratore a cassette. Il nastro magnetico costituisce il supporto delle informazioni; il registratore è lo strumento per la registrazione e la successiva lettura.
I segnali che provengono dall'unità centrale vengono convertiti, da una opportuna interfaccia (contenuta normalmente all'interno dell'unità stessa) in segnali acustici: un segnale di una certa frequenza per lo 0, un segnale di frequenza diversa per l'1. La registrazione avviene in maniera analoga alla registrazione di un brano musicale o del parlato: in effetti, in genere è possibile utilizzare un comune registratore audio, lo stesso che fino al giorno prima si usava esclusivamente per ascoltare gli ultimi successi della "hit parade". Anche le cassette possono essere le stesse cassette audio, ma è meglio usare cassette apposite, con nastro di qualità opportuna. I requisiti del nastro sono infatti un po' diversi: per registrare informazioni non è necessaria una risposta in frequenza eccezionale, mentre è molto importante l'omogeneità del nastro. La perdita di una piccola informazione, nel caso di un brano musicale, in genere non viene quasi neanche avvertita, e comunque non nuoce all'intelligibilità complessiva del brano stesso. Nel caso di una successione di cifre binarie, invece, la perdita di un solo 0 (o di un solo 1) può rendere tutto incomprensibile. Se, per esempio, l'errore si verificasse in un punto nevralgico di un programma, magari trasformando uno 0 in un 1, il programma potrebbe apparentemente funzionare ancora, ma magari dare risultati del tutto errati (e magari in un modo difficilmente identificabile).
Piú affidabili, rispetto ai registratori audio, sono i registratori "dedicati", costruiti appositamente per l'uso con un computer, e i registratori di tipo "digitale". Alcuni modelli di computer richiedono registratori particolari, e non funzionano con altri: è il caso del Commodore 64, che richiede il registratore prodotto dalla stessa Commodore e non funziona con altri (fatta eccezione per modelli costruiti da altre case sulla falsariga di quello della Commodore).
Alcuni calcolatori portatili, per comodità, incorporano già una unità a cassette: in questi casi si tratta di unità studiate appositamente e, per motivi di compattezza, fanno uso non di cassette del formato "audio", ma di microcassette.
Il registratore ha due soli pregi: il costo e le dimensioni. Per il resto è un dispositivo di memoria di massa lentissimo e poco affidabile. Il nastro è un supporto di memoria di tipo sequenziale: le informazioni debbono per forza essere disposte una in fila all'altra e non si può saltare da questa a quella. Basta pensare alle musicassette: se si vuole ascoltare l'ultimo brano della facciata, non esiste altro sistema che far scorrere il nastro fino al punto in cui si trova il brano cercato. Se si sa già dove andare a cercare, bastano pochi minuti: se non lo si sa, la ricerca può risultare penosa.
Un disco, invece, è un dispositivo "ad accesso casuale": si può abbassare la puntina su qualunque punto del disco con la stessa facilità. Andare sull'ultimo brano richiede lo stesso tempo che andare sul secondo.
Inoltre, la velocità di lettura e scrittura è limitata dalla velocità di scorrimento del nastro stesso: le informazioni non possono essere compattate al di là di un certo limite. Non è raro trovare programmi commerciali il cui caricamento da cassetta richieda magari una decina di minuti: succede anche per i giochi. È sufficiente questo elemento a farne un dispositivo del tutto inadeguato per usi professionali. Lo stesso programma su disco potrebbe richiedere pochi secondi, per il caricamento. L'unico caso in cui il registratore può rimanere una alternativa valida anche per usi professionali è quello di ambienti rischiosi o ostili per le apparecchiature piú delicate: in una fabbrica dove abbonda la polvere, un drive per dischi flessibili potrebbe funzionare magari solo con difficoltà, mentre un registratore potrebbe dimostrarsi piú affidabile. Anche per un reporter che deve viaggiare molto un registratore può essere piú comodo e meno delicato di una unità a dischi: oltre a memorizzare dati e programmi, può sempre servire per un'intervista!

Un altro tipo di supporto di memoria usato soprattutto con i computer piú economici sono le cartucce (*cartridge*): si tratta in sostanza di chip di ROM in un involucro di plastica che funge da protezione e alloggia la contattiera per il collegamento alla macchina. Per poter usare una cartuccia, il computer deve essere dotato di una apposita fessura in cui la cartuccia deve essere inserita, in modo da andare a collegarsi, tramite la sua contattiera, ai circuiti della macchina. Le cartucce contengono ROM: sono utilizzabili cioè solo in lettura. Costituiscono un buon metodo di commercializzazione di software per le macchine piú piccole, perché alleviano di molto il problema dei tempi di caricamento dei programmi (che sono disponibili virtualmente subito appena introdotta la cartuccia) ed evitano anche un altro problema, che preoccupa ovviamente molto i produttori di software. Le cartucce, cioè, non sono copiabili (o lo sono solo con strumenti molto raffinati), mentre le cassette sono copiabili molto facilmente: non ci sono problemi fisici, e le protezioni "software" che possono essere messe in pratica con questo tipo di supporto non sono molto robuste.
Le cartucce sono tuttavia costose, molto piú costose dei nastri, e nonostante la loro comodità non sono molto diffuse.

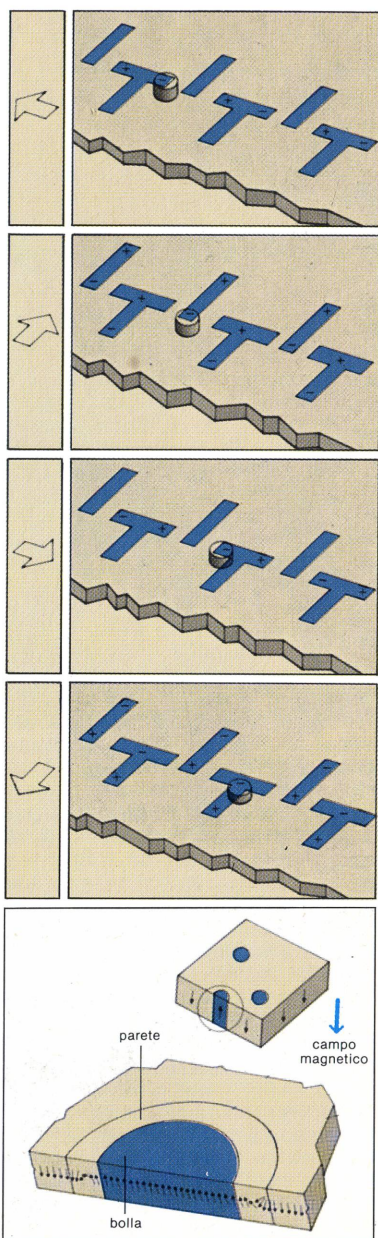
Le memorie di massa: le unità a dischi flessibili. I *floppy disk* o dischi flessibili sono i supporti di memoria secondaria piú comuni per i calcolatori personali: sono dischi di plastica realizzati in materiale di plastica flessibile (Mylar) rivestito con uno strato di ossido che gli conferisce le proprietà di magnetizzazione. I floppy disk sono grosso modo paragonabili ai 45 giri musicali, ma il materiale è molto piú flessibile. Il disco stesso è racchiuso in un involucro di plastica, per motivi di protezione, nel quale sono praticate alcune aperture per il posizionamento all'interno dell'unità di lettura/scrittura, per permettere le operazioni di accesso alla testina, per la eventuale protezione contro la scrittura.
L'unità di lettura/scrittura di dischetti flessibili (*drive* o *driver* nella terminologia inglese) è paragonabile a un giradischi, con caratteristiche tuttavia molto piú raffinate. La testina svolge la stessa funzione della testina di un giradischi: in scrittura trasforma segnali elettrici in campi magnetici, che provocano la magnetizzazione di areole sul disco; in lettura esegue l'operazione inversa. La testina (con il braccio che la sorregge) continua a muoversi, durante il funzionamento, avanti e indietro sul disco, con il quale peraltro non entra mai in contatto: "vola" al di sopra della sua superficie, a una distanza dell'ordine dei millesimi di millimetro. Il disco viene fatto ruotare a 200 giri al minuto.
Esternamente, un drive per dischetti flessibili non presenta caratteristiche peculiari: è solo un parallelepipedo chiuso, con una fessura in cui si deve inserire il disco e un meccanismo di chiusura, che al contempo impedisce la fuoriuscita del disco e abbassa la testina di lettura/scrittura.
In un drive restano numerose le parti meccaniche: l'elettronica di controllo ha lo stesso grado di affidabilità (elevato) delle parti elettroniche del computer, ma le parti meccaniche sono soggette allo stesso tipo di problemi e allo stesso tipo di usura di qualunque prodotto meccanico, anche se il livello tecnologico in gioco nella loro costruzione è elevatissimo. I drive sono comunque fra i componenti piú soggetti a difetti dovuti all'uso. Va detto comunque che, poiché la testina non entra in contatto con il disco, non ci sono problemi di usura analoghi a quelli dei dischi e dei giradischi musicali: un floppy disk non si rovina con l'uso, anche se con il tempo può perdere le sue proprietà magnetiche, ed è soggetto a danneggiamenti ad opera di agenti esterni come la polvere, le gocce di caffè o il caldo intenso.

Sui dischi le informazioni sono registrate (sempre sotto controllo del computer) in un modo un po' diverso rispetto a quello tipico per i dischi audio tradizionali. Quando si acquista un disco nuovo, prima di poterlo usare effettivamente, lo si deve sottoporre a una operazio-

I dischi flessibili nel loro drive ruotano in genere alla velocità di 300 giri al minuto; in un disco rigido la velocità è molto piú elevata, e, grazie alla velocità di rotazione, fra disco e testina si forma un "cuscinetto" di aria sul quale la testina (1) "vola". Lo spessore del cuscinetto d'aria è dell'ordine di piccole frazioni di millimetro e al confronto, come si può vedere dal disegno, una particella di fumo (2), un'impronta digitale (3), un granello di polvere (4) o un capello (5) risultano davvero enormi. Per questo, per evitare ogni possibile danno, il disco rigido viene sigillato nel suo contenitore. I dischi rigidi (fatti di alluminio rivestito di ossidi magnetici) nel contenitore possono essere anche piú d'uno (e altrettante sono allora le testine). Nella pagina a fronte, i disegni mostrano la formazione e la struttura di una bolla (in basso) e lo spostamento della bolla lungo il substrato mediante l'applicazione di un campo magnetico rotante indicato dalla freccia.
I dispositivi di memoria a bolle magnetiche non hanno raggiunto ancora lo stadio della produzione commerciale su larga scala.

ne di "formattamento": lo si deve cioè inserire nel drive e si deve far girare un programma che provvede a organizzare la sua superficie in un modo particolare. Il formato è registrato a sua volta in forma magnetica, pertanto il risultato non è qualcosa di visibile: estraendo il

disco, apparentemente sarà tutto come prima. Il modo preciso in cui il disco viene "formattato" dipende dalle caratteristiche della singola macchina, e in particolare delle caratteristiche del suo sistema operativo (il software che presiede a tutte le sue operazioni e gestisce tutte le sue risorse). Questo significa che un dischetto formattato da una macchina non può essere usato, in linea di massima, da una macchina diversa, anche se magari fisicamente i relativi drive sono esattamente identici (cosa non difficile, considerando che i costruttori di drive sono pochi e in genere vendono i loro prodotti a piú produttori di computer). Nulla di simile a quel che accade nel mondo dell'alta fedeltà, dove l'ultimo disco dei Police può essere ascoltato su qualunque "piatto" di qualunque marca. Per continuare nel confronto negativo, mentre nel campo dell'hi-fi qualunque piatto può essere inserito in un complesso con qualunque amplificatore, nel mondo dei personal computer non è possibile abbinare pari pari qualunque drive a qualunque unità centrale. I due elementi debbono essere fra loro compatibili: tutti i produttori di personal forniscono anche drive per le loro macchina (drive in genere costruiti da altri, ma adattati), e la scelta piú ovvia è affidarsi ai prodotti offerti da un'unica azienda. C'è chi fornisce i drive già incorporati nello stesso contenitore dell'unità centrale, e il problema non si pone neppure; ma c'è chi lascia libero l'utente e produce unità separate, permettendo cosí eventualmente una scelta fra i prodotti anche di altre aziende. Bisogna comunque fare attenzione e non accostare a caso.

La struttura che il formattamento produce sui dischetti è a tracce concentriche e a settori trasversali: il numero delle tracce e il numero dei settori sono fra i fattori variabili. Esistono anche dischi in cui la divisione in settori è determinata fisicamente, anziché via software, con una serie di forellini intorno al foro centrale del disco stesso: si dice in questo caso che il disco è *hard-sectored*; nell'altro caso il disco è *soft-sectored*.

Esistono tre tipi di floppy disk, sotto il

profilo delle dimensioni: con diametro di 8 pollici, di 5 pollici e 1/4, di 3 pollici (questi ultimi sono detti anche microfloppy). La maggior parte delle macchine sono predisposte per lavorare con uno dei tre tipi: il tipo da 5 pollici è il piú comune, quello da 3 pollici è il piú recente, ed è probabilmente destinato a guadagnare spazio nel futuro prossimo, grazie alle sue caratteristiche di estrema maneggiabilità. Le dimensioni in sé sono peraltro poco significative: il progressivo perfezionamento delle tecnologie costruttive ha permesso di ottenere con i dischetti da 5 pollici le stesse capacità di immagazzinamento che erano possibili pochi anni fa solo con quelli da 8, e quelli da 3 sono nati già con capacità di immagazzinamento confrontabili con quelle dei dischetti da 5 pollici: questo non è dunque un fattore molto importante.
Esistono dischi a una e a due facce: sono registrabili da una sola parte, oppure da tutte e due. Nei dischetti a doppia faccia si possono immagazzinare quantità doppie di informazioni: tuttavia l'utilità è limitata se il drive usato non è del tipo "doppia faccia". Un drive doppia faccia è semplicemente un drive che possiede due testine (e due bracci) montati in opposizione: che restano uno sopra e uno sotto il disco, e leggono quindi contemporaneamente le due facce. Se il drive è singola faccia, è sempre possibile utilizzare dischetti doppia faccia, ma bisogna fisicamente estrarli e girarli, per sfruttarli da ambedue le parti. Il vantaggio pratico è minore (ma in genere un disco doppia faccia costa meno di due dischi a singola faccia e ovviamente occupa meno spazio in casa di due dischi).
Un'altra caratteristica importante è la densità di memorizzazione delle informazioni, cioè il numero di bit che possono essere immagazzinati nell'unità di spazio. Commercialmente, esistono dischi a densità singola, doppia e anche quadrupla: il significato dei termini è ovvio. Un disco a doppia densità può ospitare una quantità di informazioni doppia rispetto a un disco a densità singola.
In assoluto, la quantità di informazioni che può essere immagazzinata su un dischetto varia da 100K circa a oltre 1000K (grandezza che si identifica anche con il nome di "1 Megabyte" o, piú sinteticamente, 1Mbyte). Questa grandezza dipende comunque non solo dalla costituzione del dischetto, ma anche e soprattutto dalla costruzione del drive: maggiore è il numero di bit che deve essere memorizzato, a parità di area, maggiore deve essere la "finezza" del dispositivo di lettura e scrittura, che deve poter determinare fra le diverse celle. Fra le macchine della fascia media, per dare un'idea, sono abbastanza comuni drive per dischi a doppia faccia e doppia densità da 360K (complessivi). È questo il tipo di drive installato sui personal computer IBM, per esempio.

Le memorie di massa: i dischi rigidi.
Una unità a dischetti flessibili è molto piú veloce di un registratore a cassette, ma è ancora molto meno veloce della RAM interna; e un dischetto può ospitare una grande quantità di informazioni, ma non ancora abbastanza grande per lavori di una certa complessità, come sono inevitabilmente la gestione di un archivio di dati o le pratiche di contabilità e, in generale, amministrative di una piccola azienda. È sempre possibile depositare le informazioni su piú dischetti, ma i dischetti vanno tolti e rimessi nel drive a mano, con grande perdita di tempo: per trattare grandi quantità di dati è necessaria una soluzione meno empirica e piú consona al principio dell'automazione.
Per realizzare memorie di massa piú capaci e dall'accesso piú veloce rispetto a quelle ottenibili con un sistema a dischi flessibili, esistono le unità a dischi rigidi. In queste unità i dischi sono di alluminio, non piú di plastica, e il rivestimento di ossido è qualitativamente migliore; il disco rigido ruota a velocità piú elevata, e la testina di lettura/scrittura vola a una distanza ancora piú ravvicinata rispetto alla superficie del disco. La qualità delle parti meccaniche in movimento deve essere altissima, ma un sistema del genere sarebbe sensibilissimo alle influenze esterne: per questo i

Un videodisco o disco ottico. Un supporto di memoria che permette l'archiviazione di informazioni sia digitali, sia analogiche. Quello nella fotografia contiene oltre centomila immagini televisive a colori.

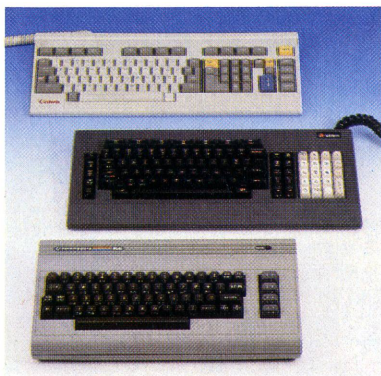
dischi rigidi sono sigillati, con tutto il complesso di lettura/scrittura e i componenti meccanici ed elettronici, in un contenitore ermetico.
I dischi rigidi sono diffusi soprattutto per i grandi sistemi, e possono raggiungere capacità anche di 600 Megabyte, ma nel corso degli anni settanta è stata sviluppata una tecnologia che permette la realizzazione di unità a disco rigido a costi relativamente contenuti e di capacità un po' minori, e che ha portato anche questi dispositivi alla portata di sistemi di uso personale. La tecnologia (nata nei laboratori IBM) è denominata Winchester, con un nome un po' fantasioso: la prima unità di questo tipo, infatti, era stata battezzata con il numero 3030, che identifica anche un famoso modello di fucile Winchester. Un disco winchester può ospitare 5, 10 Megabyte e anche piú.
In assoluto un'unità a dischi rigidi costa nettamente piú di una unità a dischi flessibili, ma i prezzi sono sempre in progressiva diminuzione; in senso relativo, però, un'unità winchester costa meno di un'unità a dischi flessibili, in termini di lire per bit di informazione memorizzabile.
Il disco rigido ha uno svantaggio: essendo sigillato all'interno del suo contenitore, non è estraibile e non è sostituibile come un floppy disk. Per raddoppiare la capacità, quindi, non è sufficiente avere due dischi, ma è necessario avere pro-

prio due unità complete. Inoltre, se succede qualcosa al disco rigido il danno è elevatissimo: con i dischetti flessibili è facilissimo fare copia di tutte le informazioni archiviate e tutelarsi contro ogni inconveniente, ma con il disco rigido no. Per questo in genere le unità a disco rigido debbono essere comunque accoppiate a un'unità a dischetti flessibili (sui quali si conservano copie di riserva del materiale memorizzato) o a qualche altra unità di riserva. Spesso la stessa unità winchester viene fornita già dotata di una unità a nastro di *backup* (cioè di riserva di sicurezza). Il nastro può essere molto lungo, e conservare anche tutto il materiale contenuto sul disco: sarebbe molto lento, se venisse usato come unità principale, ma come riserva viene normalmente usato solo in fase di memorizzazione. In caso di perdita totale del materiale sul disco rigido, però, permette la ricostruzione totale degli archivi presenti.

Altre forme di memoria di massa. Nastro, dischi flessibili e dischi rigidi sono le forme comuni di memoria di massa; altre tecnologie si sono presentate alla ribalta, ma non hanno raggiunto un grado di diffusione soddisfacente.
Interessante è la tecnologia delle memorie a bolle magnetiche, una tecnologia di tipo microelettronico che fornisce dispositivi piú veloci di una RAM, ma di natura simile a quest'ultima, con la particolarità però di conservare le informazioni anche in assenza di alimentazione. Il supporto di una memoria a bolle è uno strato di materiale artificiale disposto a sandwich fra magneti permanenti: le bolle sono piccoli campi magnetici nel materiale, e le cifre binarie sono identificate con le due polarità che questi campi possono assumere. Le memorie a bolle non hanno riscosso un grande successo: rimangono dispositivi rari e costosi, abbastanza difficili da interfacciare con un calcolatore.
Una tecnologia che sta muovendo solo ora i primi passi, ma che promette di diventare un'alternativa seria e significativa in un futuro non molto lontano, è invece quella del videodisco digitale. Le applicazioni al campo dei computer sono ancora allo stadio iniziale, ma i vantaggi possibili sarebbero enormi: grandi capacità di immagazzinamento su dischi di dimensioni standard, possibilità di integrare progressivamente in un unico supporto tutti i tipi di informazioni (dati, immagini, suoni), centralizzando il controllo di tutti i mezzi di comunicazione domestici finora separati, semplicità d'uso, affidabilità.

I dispositivi di ingresso

L'unità centrale deve comunicare con il mondo esterno: è necessario, tanto per cominciare, qualche dispositivo che permetta all'utente di inviare i suoi comandi e tutte le informazioni che vuole elaborare. I dispositivi di input possono essere di moltissimi tipi diversi: con strumenti adeguati, si può immettere nel calcolatore qualsiasi genere di informazione. Si potrebbe pensare, per esempio, di elaborare suoni musicali prodotti con metodi tradizionali: uno strumento musicale come una chitarra, per esempio. Il suono della chitarra può essere captato da un microfono, che funge da trasduttore: trasforma i segnali acustici (onde di pressione) in segnali elettrici (variazioni di tensione). Il segnale elettrico è un segnale analogico:

Una tastiera integrata all'unità di elaborazione (in basso) e due tastiere separate, collegate con un cavo a spirale.

le variazioni di tensione sono proporzionali alle variazioni di pressione acustica, e avvengono con continuità. I segnali che devono arrivare al calcolatore debbono essere, invece, segnali discreti, digitali. Il dispositivo di ingresso, in questo caso, deve essere costituito allora dall'abbinamento di un microfono con un dispositivo che trasformi segnali analogici in segnali digitali: un convertitore analogico/digitale, che fisicamente può essere realizzato con una scheda di circuiti integrati. Una volta che le informazioni sono state trasformate in segnali digitali, possono essere elaborate (con un software opportuno) esattamente come qualunque altro tipo di informazioni. I segnali digitali elaborati, poi, possono essere ritrasformati in segnali elettrici continui (analogici), simili a quelli che escono dalla testina di un giradischi o di un registratore. A quel punto un sistema amplificatore-casse acustiche potrà restituire il suono elaborato: e, se siamo stati bravi, il suono di chitarra potrebbe essere diventato piú simile a un suono d'organo, o magari di xilofono!

Con principi analoghi, si possono realizzare unità di input che permettano di comunicare al calcolatore informazioni all'origine di tipo analogico, raccolte direttamente da qualche tipo di trasduttore: un sensore di temperatura o di pressione, un analizzatore di qualche genere. In questo modo, si può pensare di usare il computer per elaborare informazioni relative, per esempio, a un processo industriale, raccolte direttamente. (È il primo passo per creare un sistema di controllo, magari un sistema d'allarme per la casa.)

I dispositivi di input piú comuni, tuttavia, per un personal computer sono un po' diversi: tastiere, tavolette digitalizzatrici, mouse.

I dispositivi di ingresso: la tastiera. Una tastiera, simile a quella delle macchine per scrivere, è il dispositivo di input piú comune: tutti i calcolatori domestici o personali ne sono dotati "di serie". A volte fa tutt'uno con il contenitore che ospita l'unità centrale e i drive; a volte è separata, e rimane collegata mediante un cavo a telefono. Nelle macchine di tipo domestico, di dimensioni piú piccole, un unico contenitore alloggia la tastiera e, al di sotto, l'elettronica dell'unità centrale; la tastiera è quasi sempre separata nei computer con ambizioni professionali; esistono anche macchine che integrano in un unico corpo tastiera, video, unità centrale e drive per floppy disk (è il caso, per esempio, di alcuni modelli della TRS-80, di alcune serie maggiori della Commodore). Queste varietà di configurazione sono puramente esteriori: possono risultare piú o meno comode a seconda della situazione d'uso, ma non sono rappresentative sotto l'aspetto qualitativo.

Per un uso professionale, tuttavia, la tastiera separata può risultare estremamente comoda: può essere tenuta nella posizione piú conveniente, può essere addirittura appoggiata sulle ginocchia. Per chi deve ricopiare testi, o deve tenere accanto del materiale di consultazione, per un traduttore che deve poter vedere bene anche il testo da tradurre, la tastiera separata è quasi essenziale. Su una scrivania inevitabilmente ingombra di mille altre cose, la tastiera separata permette di disporre la parte piú ingombrante della macchina nella posizione meno fastidiosa, mantenendo una elevata facilità di scrittura. Tuttavia può essere anche una questione di gusto personale.

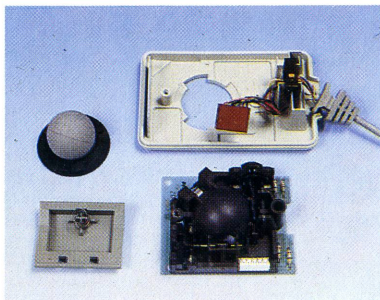
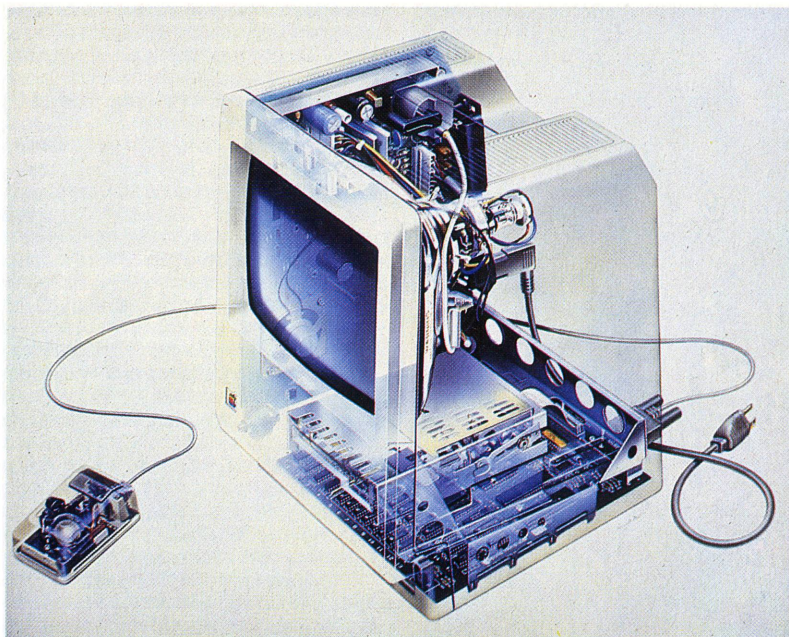
Piú importanti sono altri fattori: la qualità dei tasti, per incominciare. Le macchine piú economiche hanno a volte tasti piccoli, addirittura simili a bottoni (i primi Sinclair, per esempio lo Spectrum). Per un uso domestico e amatoriale sono accettabili, ma sono faticosi e del tutto inadatti per svolgere un lavoro pesante. Inoltre hanno una corsa in genere inferiore a quella dei tasti normali, e il tipo di digitazione possibile è molto diverso da quello che si adotta su una buona tastiera da macchina per scrivere.

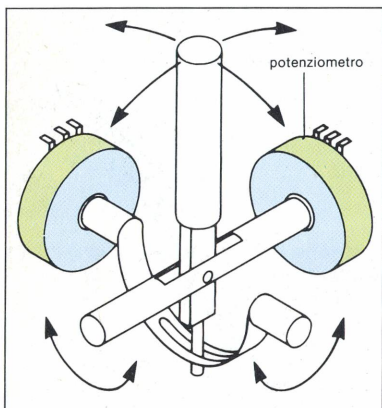
Le tastiere migliori hanno tasti a corsa normale, di dimensioni uguali a quelle dei tasti delle macchine per scrivere elettriche o elettroniche, di forma leg-

germente concava, morbidi al tocco ma anche abbastanza resistenti da dare chiara l'impressione della digitazione. È consigliabile la presenza di un "clic", che costituisce una sorta di retroazione eccellente per chi è abituato a scrivere velocemente, senza guardare molto a ciò che scrive (allo schermo, nel caso di un computer). Alcune tastiere permettono addirittura di regolare il volume del clic, al limite rendendolo pressoché inudibile (una tastiera di questo tipo è quella del Rainbow e del Professional della Digital Equipment Corporation).
Sotto il profilo costruttivo, è importante il modo in cui sono realizzati i contatti, i commutatori che sentono la pressione del tasto e inviano corrispondentemente un segnale binario all'unità centrale. I contatti meccanici sono poco costosi, ma sono facilmente soggetti a danneggiamenti per la polvere e altre cause di tipo "ambientale": hanno una vita media intorno alla decina di milioni di pressioni. I contatti capacitivi sfruttano la capacità di accoppiamento di due elementi, che viene aumentata dalla pressione del tasto; sono piú robusti e hanno una vita media che può arrivare ai 300 milioni di pressioni. I contatti a effetto Hall hanno una vita ancora superiore (20 miliardi di pressioni).

Quando si vuole utilizzare il computer per applicazioni che non siano semplicemente hobbistiche ma anche di tipo professionale, gli aspetti ergonomici assumono un grande rilievo, e la tastiera è certamente fra gli elementi piú appariscenti. La disposizione dei tasti è importante, in particolare se il computer deve essere usato spesso per l'elaborazione di testi; la disposizione "nazionale" e la disponibilità delle lettere accentate sono necessità; la presenza e la sistemazione dei tasti di controllo del cursore sono un altro fattore da tenere presente.
Il Commodore 64 (qui sopra), per esempio, ha solo due tasti per il controllo del cursore, e le altre due possibilità di movimento si ottengono premendo anche il tasto SHIFT; la tastiera dell'IBM PC (nella pagina a fronte) ha quattro tasti distinti per il controllo del cursore, sistemati in modo intuitivo a croce. I tasti di controllo del cursore dell'IBM, però, sono inseriti nel tastierino numerico e possono essere commutati come tasti numerici (il che è uno svantaggio). La qualità e la risposta dei tasti sono altrettanto importanti; per la comodità di chi lavora, una tastiera separata e inclinabile è sempre molto utile: la si può tenere lontano dall'unità centrale, rendendo piú razionale la disposizione sulla scrivania, ed è possibile addirittura scrivere appoggiandola sulle ginocchia. La presenza di tasti di funzione programmabili è un altro elemento vantaggioso (se i programmi che si usano ne prevedono l'utilizzazione o ne permettono la definizione da parte dell'utente).

Indipendentemente da tutti i fattori fisici e costruttivi, è importante poi considerare la struttura "logica" della tastiera, cioè il numero dei tasti disponibili e la razionalità o meno della loro disposizione. I tasti dei caratteri alfanumerici sono sempre presenti: la loro disposizione è quasi sempre, tuttavia, quella delle tastiere americane: per chi intende usare spesso il calcolatore per l'elaborazione di testi e magari è abituato già a usare una macchina per scrivere elettrica o elettronica, questo può costituire un handicap (non insuperabile, ma molto noioso): la W sta dove ci si aspetterebbe la Z e viceversa, dove sulle nostre macchine per scrivere c'è la M si trovano invece il punto e virgola e i due punti. I numeri sono nella posizione delle minuscole, i segni come le virgolette e le parentesi sono nella posizione delle maiuscole; e via dicendo. Per chi arriva al computer senza aver mai usato frequentemente macchine per scrivere non è un problema: alcune particolarità, anzi, possono essere addirittura vantaggiose (i numeri nella posizione delle minuscole, per esempio), e anche per la scrittura di programmi (dove difficilmente la battitura può essere veloce) non ci sono problemi.
Oltre a questi tasti tradizionali, la tastiera di un calcolatore ne possiede altri speciali, connessi al diverso tipo di funzione che ha questo dispositivo di input. Sono tasti come il CONTROL, l'ESCAPE, il RETURN (che si trova dove nelle macchine per scrivere è posto il ritorno carrello e in alcuni casi ha una funzione analoga), il tasto di cancellazione, i tasti per lo spostamento del cursore. Ci possono essere (e in genere sono molto comodi) tasti di funzione etichettati magari semplicemente F1, F2, F3 e via dicendo, programmabili: non hanno cioè una funzione in sé, ma ciascun programma può sfruttarli per facilitare determinati comandi. In un programma per l'elaborazione di testi, per esempio, un tasto di funzione potrebbe comandare la centratura di un testo o la sottolineatura; in un programma di tabellone elettronico potrebbe invece dare il comando di aggiornamento del tabellone dopo l'inserimento di nuovi dati; in un gioco potrebbe comandare il movimento verso l'alto di una nave spaziale, e via dicendo. Senza i tasti di funzione programmabili, gli stessi comandi potrebbero essere impartiti solamente mediante una serie di altri tasti, magari abbastanza lunga: è evidente quindi la comodità della presenza di un gruppo di tasti di funzione programmabili.

Può, infine, essere presente un tastierino numerico del tutto simile a quello di una calcolatrice, con le dieci cifre, i simboli di somma e sottrazione. Il tastierino numerico duplica le funzioni di altri tasti, ma risulta particolarmente comodo per coloro che devono lavorare spesso con lunghe serie di numeri, e magari sono abituati da tempo a lavorare con una calcolatrice.
Esistono tastiere su cui il tastierino numerico, oltre a questa funzione, può averne un'altra: il Personal Computer dell'IBM, per esempio, ha un tastierino con doppia funzione, numerico e di controllo del cursore (le due funzioni possono essere commutate mediante un altro tasto). Il Rainbow e il Professional della Digital Equipment Corporation, invece, hanno due tastierini, l'uno numerico e l'altro con le funzioni di controllo del cursore e altre funzioni come quelle di scorrimento del testo, utili per applicazioni particolari.
La disposizione dei tasti, al di fuori del gruppo tradizionale dei caratteri alfanumerici, non obbedisce ad alcuno standard: i tasti di controllo del cursore possono trovarsi sopra, sotto, a sinistra o a destra; possono essere due con doppia funzione (uno dà in minuscolo la riga inferiore, in maiuscolo la riga superiore; l'altro in minuscolo manda il cursore a destra, in maiuscolo a sinistra: sul Commodore 64, per esempio), quattro allineati ma separati dagli altri, oppure – soluzione piú soddisfacente – quattro disposti a croce (con interpretazione intuitiva delle quattro direzioni sopra sotto sinistra e destra). Può variare perfino la forma: i piú simpatici hanno forma triangolare, a punta di freccia (e in quattro, nella disposizione a croce, creano un quadrato), ma non è detto siano anche i piú comodi.
Anche i tasti di funzione possono essere un po' dappertutto: il Personal Computer della IBM li ha sulla sinistra, i personal Digital e Olivetti in alto, il Commodore 64 sulla destra. C'è di che divertirsi, ma anche di che impazzire, se capita di dover lavorare a breve distanza di tempo su due tastiere diverse!

Il mouse (un puntatore ottico-manuale che permette di attivare funzioni e comandi direttamente selezionati sul video, di cui si può vedere qui un modello a tre pulsanti, in uso e smontato nelle sue parti fondamentali), l'uso di comandi simbolici (icone) che raffigurano sul video la specifica funzione, l'impiego di finestre per svolgere applicazioni diverse, sono attualmente i metodi di interfacciamento piú avanzati fra l'utente e il computer. Tutti e tre sono stati introdotti per la prima volta sul mercato nel 1981 con lo Star 8010 della Xerox Corporation, un sistema potente con funzioni grafiche e di automazione d'ufficio. Applicazioni ispirate a questi concetti sono state adottate anche sui personal computer (in particolare sul Lisa e poi sul Macintosh, visibile in spaccato nella pagina a fronte, della Apple) e, in forma molto semplificata, anche su alcuni programmi per i calcolatori domestici. La semplicità e l'intuitività dell'interfaccia fra utente e macchina costituiscono una preoccupazione costante dei costruttori di hardware e dei progettisti di software, da quando l'espansione dei personal computer ha portato l'informatica a milioni di utenti senza competenze specifiche.

Attenzione, per finire, alle tastiere nazionali! Ci sono macchine vendute in America con una tastiera americana, in Italia con una tastiera "italiana", cioè con la disposizione dei tasti fondamentali come sulle classiche macchine per scrivere. In questo caso, se si dispone di tastiera italiana e ci si trova per qualunque motivo a dover usare un programma americano non italianizzato (perché lo zio è andato a New York e ci ha portato in regalo un bellissimo data base o l'ultimo grido fra i programmi di word processing), sorgeranno inevitabilmente problemi: il programma non riconoscerà i nuovi tasti, ma si comporterà come se la tastiera fosse quella americana. La soluzione migliore è forse quella di tastiere "a doppia faccia", commutabili da un modo di funzionamento all'altro, ma neanche questa è del tutto soddisfacente. Il problema è risolvibile comunque, anche se non in modo molto soddisfacente e tantomeno elegante, incollando delle "pecette" sui tasti, con i relativi simboli nella versione estranea. Per chi può permetterselo e sa di poter avere problemi di questo genere, è ottima la scelta di un personal delle case maggiori, per il quale siano reperibili sia tastiere italiane sia tastiere di altra nazionalità, e che abbia incorporato un programmino che consenta di riconfigurare il sistema in funzione del tipo di tastiera che si va di volta in volta a collegare.

I dispositivi di ingresso: il mouse. La tastiera è uno strumento dall'impostazione tradizionale, con la quale molti possono anche trovarsi a proprio agio; il mondo dell'informatica è tuttavia molto aperto alla ricerca di nuovi strumenti di lavoro, che permettono un modo piú semplice e intuitivo di interagire con la macchina. Si possono realizzare programmi che non richiedono sequenze di comandi da esprimere per esteso, ma a ogni passo offrono all'utente dei *menú*, cioè serie di opzioni alternative, fra le

In queste due pagine sono presentati altri dispositivi particolari per l'ingresso di dati. Qui sopra, alcuni tipi di joystick e lo schema del meccanismo interno di uno di questi dispositivi, con la leva che aziona i potenziometri. Il joystick viene utilizzato soprattutto nei giochi e in programmi di grafica. Nella pagina a fronte, in alto, una penna ottica (o matita luminosa) collegata a uno schermo. La penna contiene un sensore ottico che viene attivato dalla pressione contro lo schermo, rivelando il fascio di scansione: un circuito a tempo confronta il raggio con il percorso di scansione e localizza la posizione della penna. Sotto, una tavoletta sensibile alla pressione: la griglia risulta formata da due strati resistivi disposti su delle superfici membranose affacciate.

quali la scelta può essere effettuata premendo semplicemente un tasto. Integrando questa "tecnica" di programmazione con la grafica, è possibile rendere ancora piú semplice l'interfaccia verso l'utente: per mezzo di un dispositivo chiamato *mouse*, cioè letteralmente "topolino".

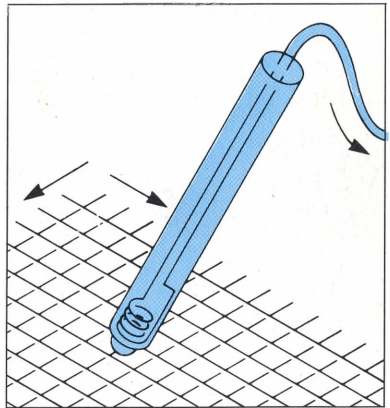
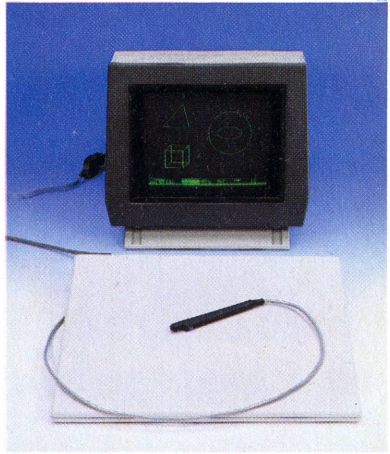
Il mouse è sostanzialmente un dispositivo di controllo del cursore: è un piccolo oggetto che si può muovere sulla scrivania e fa spostare sullo schermo il cursore in modo corrispondente. Si sposta il mouse a destra, il cursore va a destra; si sposta il mouse in avanti, il cursore sale. Il meccanismo di scelta diventa simile a quello di certi questionari statistici: "sbarrare la casella corrispondente alla risposta scelta". Il mouse infatti (si può vederne la struttura alla pagina precedente) è dotato di un pulsante che fa la stessa funzione della crocetta sulla casella: si muove il mouse fino a portare il cursore in corrispondenza dell'opzione desiderata, poi si preme il pulsante per comunicare la propria intenzione.

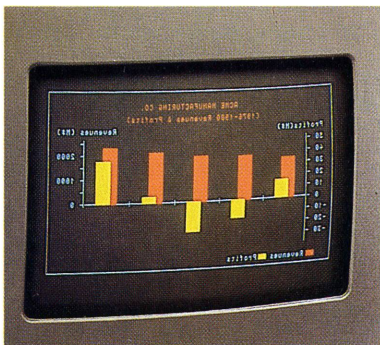
Il Macintosh della Apple è stato il primo personal computer a offrire la possibilità di un uso esteso di questo semplice strumento di input, abbinandolo a un uso altrettanto esteso della grafica: le stesse opzioni vengono presentate su questa macchina non solo con la loro descrizione a parole, ma anche mediante *icone*: come dice la parola stessa, si tratta di simboli grafici che raffigurano sinteticamente le funzioni possibili. Cosí, per esempio, il disegno di una gomma può rappresentare la funzione "cancellazione", il disegno di un cestino della carta straccia può stare per "eliminare", il disegno di un mano con una matita e un foglio indica la scelta del programma di word processing, e via di questo passo.

All'interno di un programma di grafica, il mouse può svolgere una funzione simile a quella della matita: non c'è bisogno di un comando che dica esplicitamente "traccia una linea a partire dal punto tal dei tali fino al punto tal dei tali", ma basta guidare il cursore con il mouse e definire con il pulsante i punti di inizio e fine della linea.

I dispositivi di ingresso: joystick e paddle. Il *joystick* è un dispositivo di ingresso che ha avuto una grande diffusione soprattutto nel campo dei giochi. Sostanzialmente è un'asta che può spostarsi attorno a un perno, nelle quattro direzioni, ed è munita di un pulsante che invia alla macchina un singolo impulso (e che nei classici giochi spaziali serve per "sparare"). Gli spostamenti dell'asta permettono di comunicare in forma intuitiva al computer coordinate di spostamento sullo schermo: per esempio, le coordinate della posizione in cui si vuole spostare – per restare nel campo dei giochi – la propria navicella spaziale o la racchetta da ping pong.

Il joystick può essere impiegato nello stesso modo anche in programmi applicativi piú "seri": è meno versatile, ma svolge una funzione non dissimile da quella del mouse. Le *paddles* sono dispositivi analoghi, ma basati su una coppia di potenziometri, anziché su un'asta. I due potenziometri hanno la forma in genere di manopole rotanti come quelle che servono per la regolazione del volume o dei toni su un amplificatore o della sintonia su un apparecchio radiofonico.

Mouse, joystick e paddles sono dispositivi di input caratterizzati fondamentalmente da un modo d'uso molto intuitivo, ma non hanno guadagnato ancora una diffusione amplissima. È importante ricordare, però, che non possono essere usati a piacere: possono essere usati solo se il programma è stato scritto esplicitamente per questi dispositivi.

I dispositivi di ingresso: penne ottiche e tavolette grafiche. Penne ottiche e tavolette grafiche sono dispositivi di input usati in particolare nel campo della grafica: quello che con un mouse si fa sulla scrivania, si può fare direttamente con una penna ottica sullo schermo. La penna ottica o matita luminosa è un dispositivo dalla forma simile a quella di una matita, in grado di emettere un sottile raggio luminoso, che può essere "sentito" dallo schermo. È molto usata su sistemi dedicati, per esempio per il disegno tecnico.

La tavoletta grafica è un dispositivo di digitalizzazione, basato sul principio di cui abbiamo parlato in precedenza: il disegno viene tracciato sulla superficie della tavoletta con un apposito stilo come con una matita. Al di sotto della superficie della tavoletta una griglia di fili definisce il sistema di coordinate: la risoluzione è variabile da sistema a sistema. Quando lo stilo passa all'incrocio fra un filo orizzontale e uno verticale, viene registrato il suo passaggio: il punto è definito univocamente dai due fili (il principio è del tutto analogo a quello

delle coordinate cartesiane nell'algebra). Le tavolette grafiche, poi, sono dotate di una serie di commutatori che possono essere attivati con lo stesso stilo, e che comunicano al calcolatore il tipo di regola su cui si deve basare nell'interpretazione dei dati che gli arrivano: si può cosí precisare l'intenzione di disegnare un triangolo piuttosto che magari un cerchio. Il disegno a mano libera, a questo punto, può essere anche approssimativo, perché pensa il calcolatore a precisarlo, sulla base delle regole che gli sono state comunicate.
La tavoletta grafica è una forma relativamente semplice di digitalizzazione: trasforma grandezze analogiche come i nostri disegni continui in grandezze digitali (rappresentate mediante numeri discreti). Esistono tavolette grafiche economiche, adatte per l'uso con piccoli calcolatori personali, per sperimentare nel campo della *computer graphics*; esistono anche grandi digitalizzatori per usi professionali, dotati di dispositivi speciali anche per la "lettura" di disegni realizzati con metodi tradizionali. Per esempio, un complesso disegno d'architettura può essere disposto al di sopra della superficie di un digitalizzatore ed esplorato completamente con l'apposito dispositivo, che "riconosce" i tratti del disegno e ne permette la trasformazione in segnali digitali, memorizzabili ed elaborabili dal calcolatore.

I dispositivi di uscita

I modi in cui si può pensare di far interagire il calcolatore con il mondo esterno sono vari quanto i modi in cui si può pensare di comunicare informazioni dall'esterno alla macchina. Si tratta sostanzialmente di trasformare i risultati

Confusione nei 2 sensi	Confusione in 1 senso
O e Q	C letto come G
T e Y	D letto come B
S e 5	H letto come M o N
I e L	J, T letti come I
X e K	K letto come R
I e 1	B letto come R, S o 8

Per valutare un monitor non è necessario tener conto solo delle sue doti estetiche e della sua capacità risolutiva, ma anche di una serie complessa di fattori ergonomici. La presentazione a lettere chiare su fondo scuro è preferita in genere per causare il minor disturbo alla vista, ma alcuni costruttori sono riusciti a superare ormai la maggior parte degli svantaggi di una presentazione su fondo chiaro con lettere scure. Gli schermi antiriflesso sono quasi standard, ma raramente invece i costruttori forniscono supporti che permettano all'utente di orientare il monitor in funzione delle proprie esigenze. Sopra, alcuni tipi di confusione possibili nell'identificazione di simboli, alla lettura di testo su un monitor di qualità non soddisfacente.

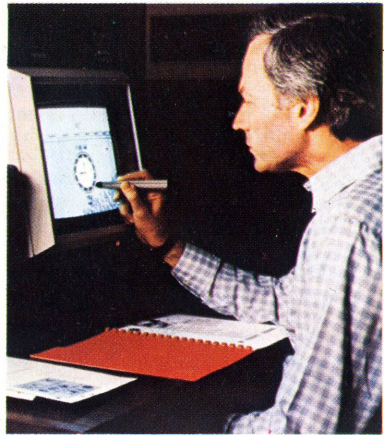
dell'elaborazione, che all'interno del computer sono in forma digitale – successioni di cifre binarie –, in forme adatte per la comprensione da parte di un essere umano o per l'utilizzazione in contesti particolari.

Le unità di uscita: il video. Il video è l'unità di uscita piú comune, corredo tradizionale dei calcolatori come la tastiera. Nel caso dei calcolatori domestici della fascia di prezzo inferiore, solitamente il video non è compreso nella configurazione di base fornita dal costruttore: sono pensati in genere per l'uso con il televisore domestico. Il televi-

sore è piú che adeguato, in effetti, per un uso limitato e poco impegnativo (ed è una soluzione eccellente per ridurre l'investimento iniziale). È sufficiente collegare il computer, con l'apposito cavetto, alla presa d'antenna del televisore (praticamente tutti i televisori di recente costruzione hanno già una presa adatta): bisogna trovar la sintonizzazione opportuna, ma è sufficiente riservare un canale per il computer e l'operazione può essere compiuta una volta per tutte. (È come sintonizzarsi su una emittente televisiva.)

Molti calcolatori hanno il video integrato con l'unità centrale (ed eventualmente i drive per i dischetti): non ci sono problemi di scelta. I video integrati sono studiati specificamente per la macchina e funzionano sicuramente bene. Può essere interessante, tuttavia, che la macchina disponga di un'uscita per il

Sotto, una griglia per preparare il disegno di animazione visibile sullo schermo. L'esempio si riferisce alla programmazione del Commodore 64: la figura è uno "sprite" che, una volta costruito, può venire utilizzato come un oggetto unico. La sua costruzione e la sua attivazione comportano il calcolo di codici numerici come quello rappresentato qui sopra.

Alcuni sistemi permettono di utilizzare, per l'ingresso dei dati, in modo simile alla penna ottica, direttamente il dito: sono i sistemi a "touch sensitive screen", ovvero a schermo sensibile al tocco. Il metodo è usato soprattutto su sistemi medio-grandi orientati alla grafica o al controllo di processo; il primo personal computer con uno schermo di questo tipo è stato prodotto dalla Hewlett-Packard.

collegamento di un video esterno, per qualunque esigenza.

Le unità video dedicate sono fondamentalmente analoghe a televisori privi dell'elettronica per la ricezione di trasmissioni televisive: tuttavia in genere hanno una risoluzione maggiore e permettono una visualizzazione di qualità piú elevata. Queste unità prendono il nome di *monitor*.

La prima distinzione da tracciare è fra

monitor monocromatici e monitor a colori. La distinzione è di costo: un buon monitor a colori costa nettamente piú di un monitor monocromatico. È poi una distinzione di funzionalità: se l'obiettivo è esclusivamente quello di elaborare testi o di usare programmi per la gestione di archivi di dati, un monitor monocromatico è piú che sufficiente. Per tutte le applicazioni grafiche, invece, un monitor a colori è quasi indispensabile. Non c'è dubbio che, anche nella grafica per applicazioni commerciali, per esempio in un istogramma o in un diagramma a torta, una buona scelta di colori possa avere un impatto comunicativo di grande efficacia.

Lo schermo è suddiviso in molti piccoli quadratini o *pixel* (contrazione di *picture cell*, elemento d'immagine), le unità piú piccole che possono essere controllate individualmente. Il numero di pixel è dunque una misura della risoluzione possibile: tanto maggiore il numero di pixel, tanto piú fine è il livello di dettaglio raggiungibile. Qualunque immagine sullo schermo è ottenuta per punti: il che è particolarmente evidente nel caso dell'immagine di un carattere alfabetico. Ogni singolo carattere è disegnato sullo schermo all'interno di una matrice di punti di dimensione fissa: quanto maggiore è il numero di pixel dello schermo, tanto piú numerosi possono essere i punti che costituiscono la matrice dei caratteri, e ovviamente migliore sarà la visualizzazione.

Il dettaglio dei caratteri non dipende solo dalle qualità del monitor, ma anche dalle caratteristiche dell'unità centrale. Se il monitor è integrato nella configurazione base di un personal computer, le sue caratteristiche sono sicuramente corrispondenti a quelle controllabili dall'hardware (o dai programmi inscritti in forma permanente nelle ROM del sistema), perché del problema si è preoccupato all'origine il costruttore. Quando l'unità centrale è venduta separatamente, si può ricorrere a qualunque monitor, ma per molti aspetti il tipo di risposta dipenderà dalle caratteristiche dell'unità centrale e non da quelle del video stesso. Certo, se il video è a fosfori arancione, i caratteri risulteranno di questo colore: questa caratteristica dipende dalla costruzione fisica del monitor. Se però, per esempio, il computer è predisposto per visualizzare 40 caratteri per riga su 25 righe, questa è la risoluzione che si otterrà sullo schermo, per quanto raffinato possa essere il monitor. Per poter visualizzare un maggior numero di caratteri per riga è necessaria una modifica dell'unità centrale, con una sua eventuale espansione con una scheda specifica di interfaccia che metta a disposizione i mezzi per controllare meglio le risorse del monitor.

Le dimensioni dello schermo sono relativamente poco importanti: i prodotti che si trovano comunemente in commercio sono standardizzati sui 9 o sui 12 pollici. I computer trasportabili, per motivi di compattezza, sono quasi sempre dotati di schermi da 9 pollici: alcuni sono comunque di buona qualità e non fanno rimpiangere dimensioni maggiori. Per chi lavora con il computer e non ha problemi di trasporto frequente, lo schermo da 12 pollici è comunque meno faticoso.

I monitor monocromatici sono disponibili con tre tipi di fosfori (i materiali luminescenti che trasformano l'energia elettrica del fascio del cannone elettronico in energia luminosa): bianchi, gialli e verdi. Non c'è accordo universale sul colore migliore per la vista umana: la scelta è lasciata molto al gusto e alle preferenze personali.

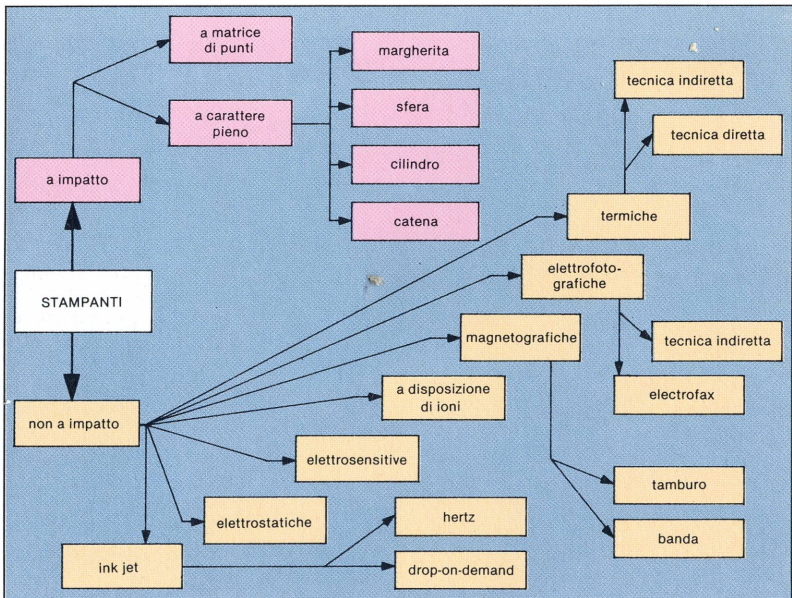
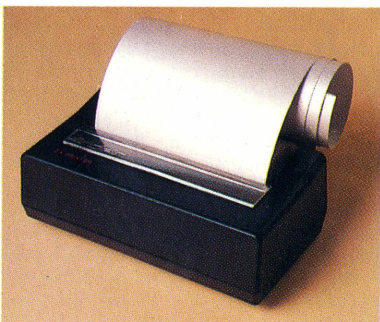
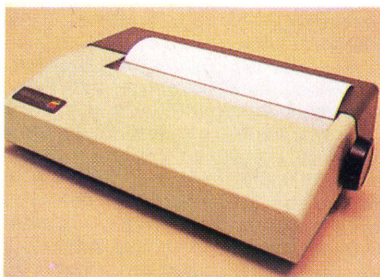
Le unità di uscita: le stampanti. Il video fornisce solo una copia "volatile" del dialogo fra l'utente e la macchina e dei risultati delle elaborazioni: in molti casi è invece desiderabile una copia di tipo permanente (una *hard copy*, nella terminologia inglese), su carta. Per questo scopo esistono le stampanti, che svolgono una funzione simile a quella di una macchina per scrivere, ma sotto controllo del calcolatore.

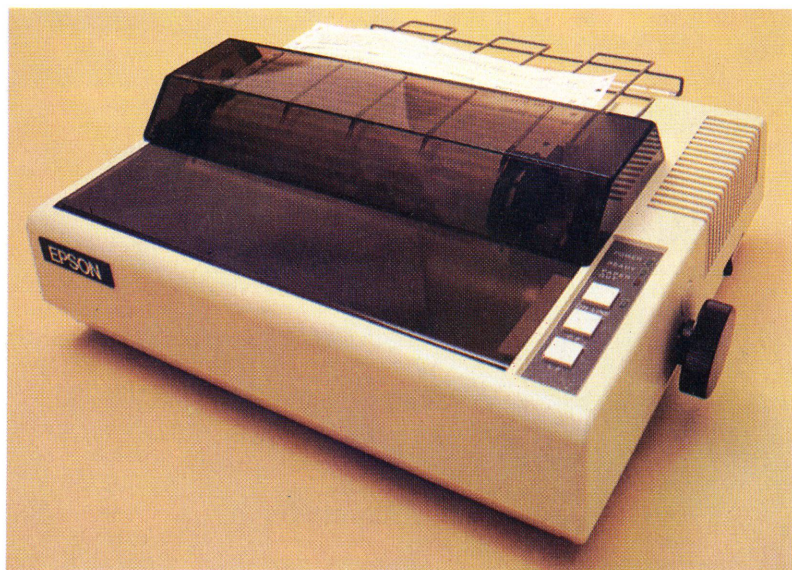
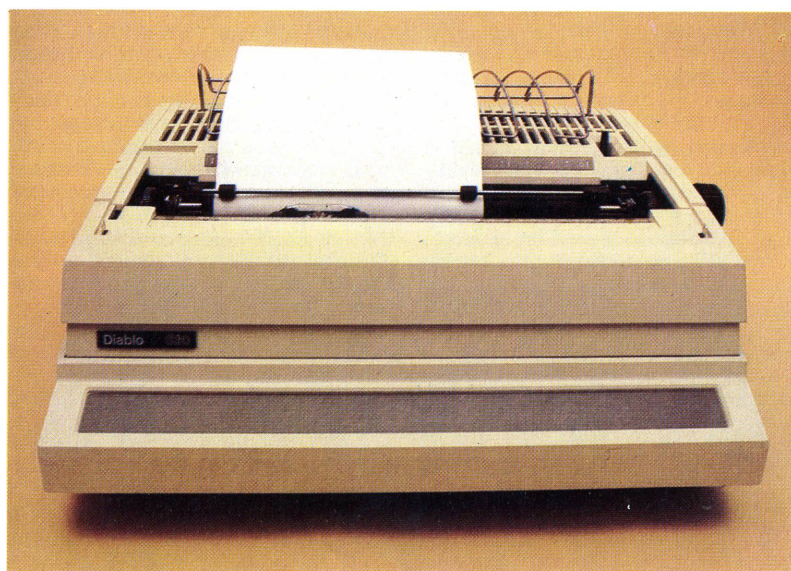
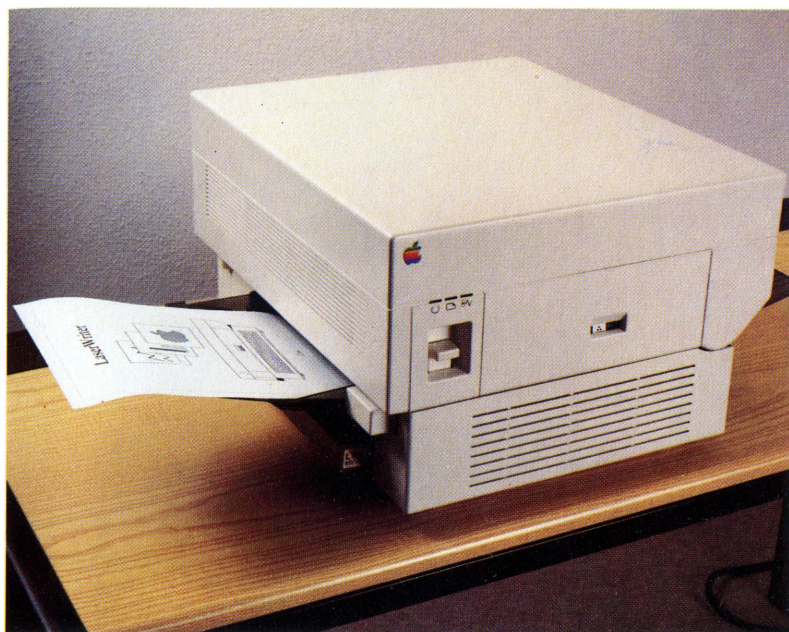
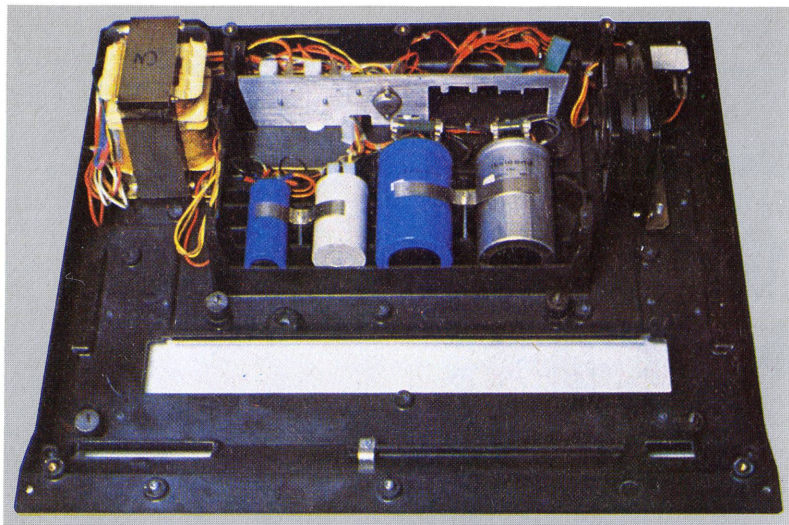
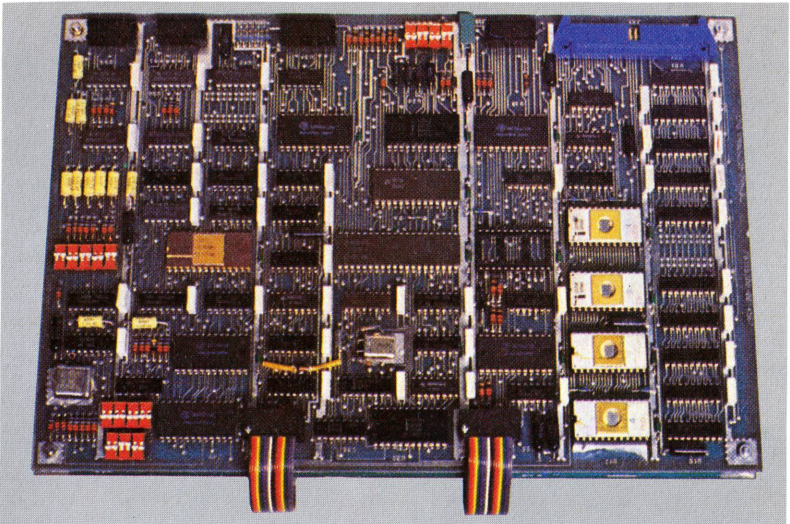
Le stampanti si possono suddividere in due grandi classi: a impatto e non a impatto. In quelle della prima classe, il meccanismo di stampa entra in contatto con la carta; nelle altre no. Questa

classificazione non ha nulla a che vedere con la qualità della stampa: si possono avere stampe di cattiva e di buona qualità con tutti e due i sistemi. La distinzione ha invece qualche riflesso sulla rumorosità: le stampanti a impatto hanno inevitabilmente parti meccaniche in movimento in piú, e risultano piú rumorose.

Fra le stampanti a impatto si può distinguere fra stampanti a carattere intero e stampanti a matrice. Nelle stampanti a carattere intero c'è un unico elemento di stampa per ciascun carattere: il funzionamento è analogo a quello delle macchine per scrivere tradizionali, dove ciascun martelletto riporta la matrice di un carattere intero. In effetti, le stampanti a impatto a carattere intero possono essere vere macchine per scrivere elettroniche adattate. Il meccanismo di stampa è del tipo a sfera (come nelle classiche macchine per scrivere IBM) o, piú frequentemente, a margherita. Le stampanti a martelletti sono ormai rare (tranne che per i sistemi di telescriventi, al di fuori delle applicazioni per computer).

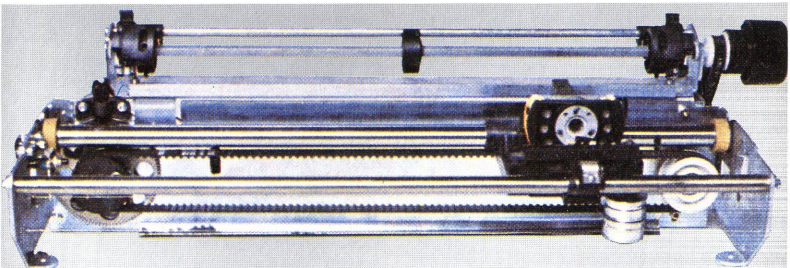
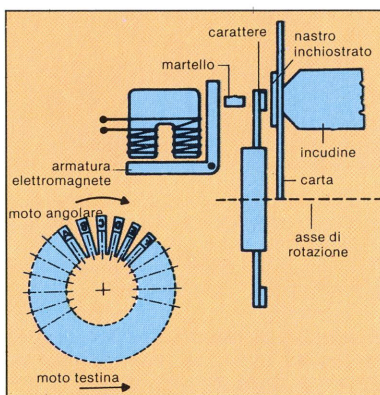
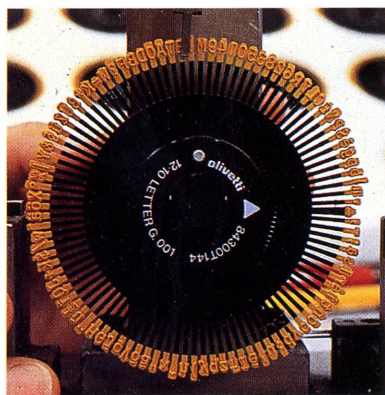
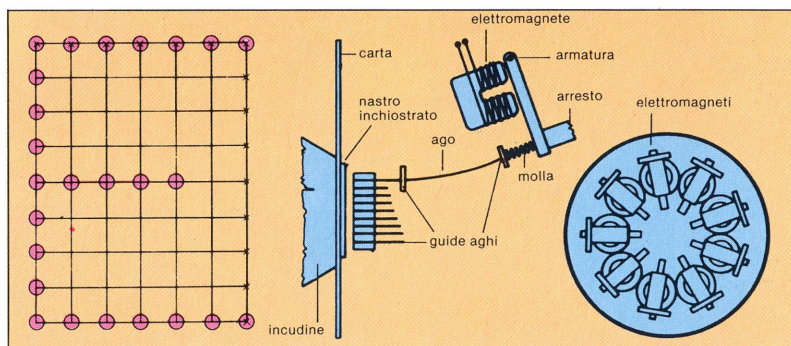
Le stampanti a matrice realizzano ciascun carattere con piú elementi di stampa, tipicamente degli aghi. Come per la rappresentazione su video, ogni carattere è disegnato entro una matrice di punti di dimensione fissa, per esempio, di 7 punti di base per 9 punti di altezza. L'elemento di stampa, in questo caso, è rappresentato da una fila di 9 aghi disposti in verticale, che vengono attivati selettivamente, da impulsi di corrente (è il solito meccanismo di 0 e 1): quando un ago è attraversato dalla corrente, si sposta in avanti e va a colpire il nastro inchiostrato, lasciando sulla carta il segno di un puntino. Sette passi successivi depongono sulla carta tutti i puntini che servono a disegnare un carattere. Il tratto del carattere ovviamente non è continuo, ma può essere ugualmente abbastanza nitido: diventa tanto migliore quanto maggiore è il numero di

Una classificazione dei vari tipi di stampanti per computer, in funzione della tecnologia impiegata.

Cinque tipi diversi di stampanti, che usano principi di stampa fra loro differenti, e che coprono l'arco dalle piú economiche alle piú costose. A sinistra in alto, una stampante termica, che usa carta speciale e ha un meccanismo di stampa ad aghi, che però non entrano in contatto diretto con la carta: attivati selettivamente, gli aghi si riscaldano tanto da indurre reazioni chimiche puntuali sulla carta. Al centro, un'altra stampante molto economica, di tipo elettrostatico: anche questa richiede carta speciale in rullo. Qui in basso, una delle piú diffuse stampanti a impatto, ad aghi a matrice di punti, che utilizza carta comune, in moduli continui o in fogli singoli: è ancora una stampante di costo contenuto, veloce ed efficiente, ma rispetto alle precedenti ha una qualità di stampa abbastanza piacevole (e, come le precedenti, ha capacità grafiche). A destra in basso, una stampante a impatto con elemento di stampa del tipo a "margherita" (del tutto analogo a quelli delle macchine per scrivere elettroniche) e quindi a carattere continuo: piú costosa e assai piú lenta, fornisce però testi di qualità elevata. L'ultima, a destra in alto, è una stampante raffinata, non a impatto: è una stampante laser, basata su un principio analogo a quello della fotocopiatrice, in grado di produrre rapidamente pagine intere, con grande qualità di stampa.

Diablo

aghi e piú ampia è quindi la matrice che può essere utilizzata nella definizione del carattere.
Le stampanti con un piccolo numero di aghi danno una stampa di bassa qualità: le ascendenti non sono vere ascendenti, per esempio, ma per poter far stare una "p" o una "g" nella matrice è necessario alzarle, e la gamba di queste lettere arriva alla base di una "a" o di una "o". La leggibilità è sufficiente, e se magari si devono scrivere solo listati di programmi con l'uso delle sole maiuscole, una stampante di questo tipo è adeguata. Per testi lunghi e impegnativi – per applicazione di word processing – invece il risultato non si può definire molto soddisfacente.
Una stampante a margherita dà risultati qualitativamente eccellenti per il word processing, ma è molto piú lenta di una stampante a matrice; per avere a disposizione caratteri diversi, in una stampante a margherita è sufficiente cambiare l'elemento di stampa; in una stampante ad aghi è necessario modificare la memoria dei caratteri (una ROM installata dalla casa costruttrice) o scrivere un programma (se la stampante lo permette) che escluda la memoria interna e consenta il pilotaggio diretto degli aghi di stampa. Tuttavia, se non è mai stata realizzata, per una certa stampante a margherita, una margherita con i caratteri cirillici, non si può scrivere un testo russo in cirillico; con una stampante a matrice controllabile via software, e con un po' di pazienza, invece, la cosa è possibile.
C'è qualcosa da spiegare. Una stampante moderna è, in un certo senso, un computer in se stessa: possiede un microprocessore che presiede al controllo di tutte le sue operazioni, una ROM (memoria a sola lettura) in cui è depositato all'origine il disegno di tutti i caratteri che può stampare, circuiti di input per la ricezione dei segnali dal calcolatore, circuiti per il pilotaggio degli aghi. Non si tratta cioè di unità periferiche grezze, ma di unità dotate di una loro "intelligenza".
Per questo sono flessibili, ma possono presentare qualche problema di collegamento con il calcolatore: i caratteri che vediamo sul video possono non corrispondere esattamente a quelli nella ROM della stampante. I caratteri, per le due macchine, sono solo dei codici, successioni di bit: esistono alcuni stan-

Le parti che compongono una stampante ad aghi: nella pagina a fronte, la base con l'alimentazione, qui sopra la scheda con tutti i circuiti elettronici e, in basso, la parte elettromeccanica.

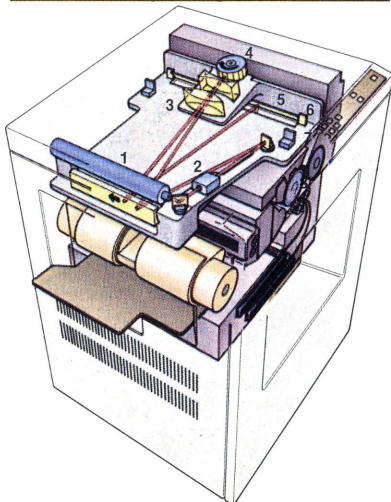
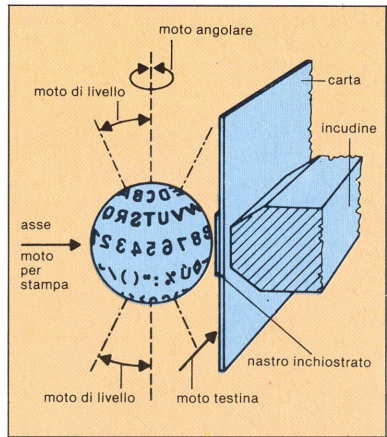
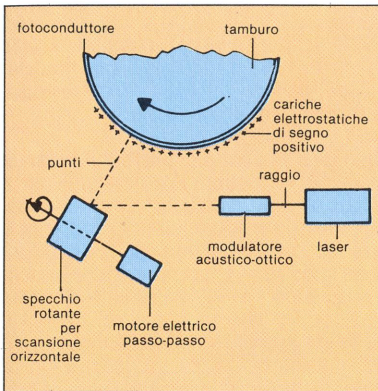
dard a cui i costruttori si attengono, in particolare il codice ASCII (American Standard Code for Information Interchange, codice standard americano per lo scambio di informazioni), che fa corrispondere a ciascun carattere una particolare successione di 0 e 1. Il codice ASCII tuttavia definisce solo i caratteri fondamentali, ma lascia aperte molte possibilità: cosí, in particolare, i codici con i valori piú alti vengono usati da macchine diverse in modo diverso, in genere per caratteri grafici, dove ogni costruttore si sbizzarrisce come meglio crede. Cosí bisogna sincerarsi di poter far colloquiare con soddisfazione la propria unità centrale con la periferica di stampa prescelta.

Queste ultime considerazioni valgono anche per le stampanti non a impatto, fra le quali sono comuni le stampanti termiche, mentre le stampanti a getto d'inchiostro e le stampanti con procedimenti di tipo xerografico a laser, qualitativamente molto superiori, sono ancora apparecchiature molto costose che non hanno raggiunto il mercato dei personal computer (ma, con tutta probabili-

tà, data la rapidità con cui le nuove tecnologie si diffondono e i costi si abbassano, sono destinate a investire anche questa fascia di mercato in tempi relativamente brevi).

Le stampanti termiche sono ad aghi, come quelle a impatto, ma gli impulsi di corrente provocano un riscaldamento selettivo degli aghi che provocano l'annerimento, nei punti corrispondenti, di una carta speciale, sensibile al calore. Queste stampanti sono poco rumorose, ma hanno il difetto di richiedere l'uso di carta speciale, e in genere non producono una stampa di qualità elevata.

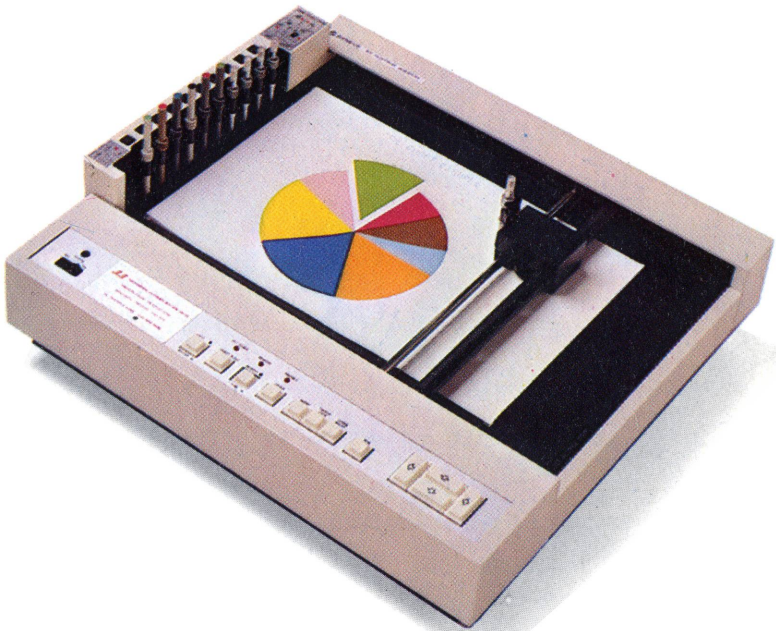
Il meccanismo della stampa a getto d'inchiostro è molto interessante: una serie di ugelli espelle gocce d'inchiostro che, in volo, passano attraverso elettrodi che le caricano elettricamente

In alto a sinistra, la matrice 7 × 9 di una media stampante ad aghi: tutti i caratteri sono disegnati per punti all'interno di questa matrice base (nel nostro caso i cerchi indicano i punti eccitati per dare la lettera E). A fianco, è schematizzato il meccanismo di stampa: i segnali in arrivo mettono in azione un meccanismo elettromagnetico che effettua la selezione degli aghi da attivare e ne causa l'impatto con il nastro inchiostrato, che va a lasciare la traccia di inchiostro sul foglio di carta. Sopra, a sinistra, una tipica margherita per stampante di alta qualità: non è sostanzialmente diversa dalle margherite delle macchine per scrivere elettroniche (in effetti, molte macchine per scrivere elettroniche possono essere dotate, o addirittura nascono ormai già dotate, di un'interfaccia che consente di utilizzarle come stampanti per computer).
La tecnica di stampa a margherita è schematizzata qui sopra.

La tecnica di stampa a sfera non è molto diversa da quella a margherita: l'elemento di stampa però ha tre ordini di moto, anziché due soli. Qui sotto, lo schema di scansione orizzontale per scrittura laser e lo schema di una stampante laser: il laser (1) invia il suo raggio attraverso un modulatore (2); il raggio, riflesso più volte, attraversa due lenti di correzione (3), poi viene riflesso su uno specchio rotante (4). Sui lati dell'ultimo specchio (5) vi sono specchietti di raccolta (6) e rilevatori di scansione (7) per la sincronizzazione. Lo specchio (5) devia il raggio sul tamburo fotoconduttore (8); l'immagine lo scarica selettivamente e viene trasferita su carta con procedimento xerografico.

e attraverso magneti che ne deflettono la traiettoria (secondo le indicazioni fornite dalla memoria che contiene il disegno dei caratteri) in modo da inviarle nel punto giusto sulla carta, oppure da farle ricadere in un pozzetto di recupero (se corrispondono a un punto che deve rimanere bianco). Anche la stampa a getto d'inchiostro è una stampa a punti, ma il meccanismo è più "morbido" della stampa ad aghi e il risultato è passibile di miglioramenti notevoli.

La stampa xerografica è analoga alla stampa di fotocopie: è una stampa a pagina, non a carattere, e sfrutta le grandi potenzialità del laser: il procedimento è complesso, e finora i prodotti di questa famiglia sono rimasti molto costosi e adatti solo per sistemi di stampa di grande qualità (per esempio, per siste-

mi di fotocomposizione professionali per editori).
Una caratteristica significativa delle stampanti è la velocità, che viene data normalmente in caratteri al secondo: i valori dichiarati sono sempre quelli massimi, che raramente vengono raggiunti. Le stampanti a impatto a carattere intero sono le piú lente: incidono molto i tempi necessari per il posizionamento e il movimento della margherita rotante. La velocità varia da poche decine di caratteri al secondo a meno di un centinaio per le macchine piú veloci: gli aumenti di velocità si pagano piuttosto cari. Le stampanti ad aghi sono molto piú veloci: superano quasi sempre il centinaio di caratteri al secondo, le piú veloci raggiungono anche i 200. Le stampanti termiche hanno velocità intermedie. Le stampanti laser sono le piú veloci e qualitativamente le migliori.
Fattori da tenere in considerazione, nella scelta di una stampante, sono anche le modalità di stampa possibile: le stampanti a margherita permettono di cambiare stile di carattere con il cambio della margherita; le stampanti ad aghi possono permettere la stampa condensata, la stampa espansa, il neretto (nella stampa condensata la matrice dei caratteri viene ridotta, in quella espansa ampliata; il neretto viene ottenuto passando due o tre volte sopra gli stessi caratteri). Sono relativamente poche le stampanti che mettano a disposizione la stampa proporzionale (in cui i caratteri non occupano tutti il medesimo spazio, ma la "i" occupa per esempio uno spazio minore delle "o" o della "m"). Stampa proporzionale, per esempio, è quella normale dei libri (questo non fa eccezione): anche nell'uscita su carta da calcolatore può dare al prodotto finale caratteristiche di maggiore raffinatezza ed eleganza.
Le stampanti a margherita sono preferite nell'ambiente d'ufficio, per la realizzazione di lettere e documenti ufficiali: il risultato è simile a quello delle tradizionali macchine per scrivere, quindi si mantiene una continuità esteriore con i procedimenti del passato, ma con maggiore efficienza e pulizia. Per l'uso strettamente personale, una buona stampante ad aghi è quasi sempre la soluzione piú convincente: anche per un professionista che lavori frequentemente con testi, anziché con informazioni di tipo numerico o con listati di programmi. Al risparmio di denaro si accompagnano una maggiore versatilità e una maggiore velocità, con una stampa che si conserva ben leggibile. Per chi può permetterselo, l'ideale può essere l'uso di due stampanti, una ad aghi, veloce ed economica, per le "bozze" dei lavori, che possono subire ulteriori correzioni e modifiche (è comodo potersi portare in giro, a volte, un pacchetto di fogli da rileggere in qualunque posto, o magari da far leggere ad altri), e una stampante a margherita, piú lenta ma con una elevata qualità di stampa, per la realizzazione dell'elaborato finale.

Le unità di uscita: il plotter. Il *plotter* è un piccolo "robot" specializzato per il tracciamento di disegni su fogli di carta, un braccio meccanico dotato di una o piú penne, comandato dal calcolatore con opportuni programmi. È lo strumento principe per la produzione di copie permanenti di disegni costruiti con programmi di grafica al calcolatore.
Non esistono, va detto, solo plotter a penna: esistono anche plotter elettrostatici, dove il meccanismo di scrittura è basato su una testina ad aghi analoga a quelle delle stampanti a matrice. In effetti, alcune stampanti a matrice permettono di stampare non solo caratteri alfanumerici, ma anche disegni: si dice allora che sono stampanti con capacità grafiche. Normalmente il modo grafico non può essere utilizzato contemporaneamente al modo caratteri alfanumerici, ma può essere attivato solo da programmi opportuni.
La testina ad aghi di un plotter elettrostatico non provoca immediatamente la stampa: gli aghi, caricati elettricamente in modo selettivo, vanno a sensibilizzare la carta. In un secondo momento viene spruzzato un liquido contenente particelle di carbone in sospensione, che vengono attirate sui punti sensibilizzati. Le particelle vengono poi essiccate in

Un plotter a piú penne colorate e a tavola piana: strumento principe per l'output grafico di un sistema personale.

un'ultima fase, durante la quale vengono anche eliminate le particelle in eccesso. Il disegno risultante è ovviamente un disegno a punti.
Nei plotter a penna il disegno non è a punti, ma a linee continue. Nei plotter a tavola piana il meccanismo di scrittura è costituito da una traversa che si muove orizzontalmente, sulla quale è disposta una torretta, che porta la penna di scrittura e si muove invece in senso verticale: la combinazione dei due movimenti permette di raggiungere con la penna qualunque punto del foglio fissato alla tavola.
Nei plotter a rullo e in quelli a tamburo, invece, non si ha traversa a movimento orizzontale, ma solo la torre portapenne: il movimento orizzontale è fornito dal rullo o dal tamburo, rispettivamente.
Nei plotter a rullo la carta viene trascinata da un opportuno meccanismo, che richiede tuttavia l'uso di carta speciale con i bordi forati; nei plotter a tamburo la carta viene fissata con altri sistemi, e può essere quindi carta qualunque.
Nel caso dei plotter a tavola piana le dimensioni del foglio sono vincolate dalle dimensioni della tavola stessa, che sono normalmente comprese fra i 25 × 38 e i 130 × 203 cm. Anche nei plotter a tamburo il supporto fissa un limite alle dimensioni massime del foglio di carta utilizzabile: normalmente 85 × 150 cm. Nei plotter a rullo, invece, è determinata solo una delle due dimensioni, variabile da 30 a 180 cm a seconda del modello; essendo la carta in rotolo, l'altra dimensione è vincolata solamente dalla lunghezza massima dei rotoli di carta forniti dalla casa costruttrice.
Il numero di penne che un plotter può utilizzare è variabile: i plotter piú semplici ne hanno una sola, i piú raffinati arri-

vano a otto penne. Quando la penna è unica, il cambiamento di colore rende necessario l'intervento dell'utente, che deve provvedere a sostituire la penna; nei plotter a piú penne, si può istruire da programma la torretta perché vada a deporre una penna e ne raccolga un'altra dalla sua posizione di riposo.
Le penne stesse possono poi essere di tipo diverso: a china, a sfera con inchiostro liquido, a sfera pressurizzate oppure con punta di feltro (tipo pennarello).
Esistono plotter elettrostatici molto economici e compatti, adatti anche all'uso domestico: la spesa è tanto contenuta da rendere pensabile l'acquisto anche solo a titolo di divertimento. Il costo di un plotter a penna è sempre piú elevato, anche se si trovano plotter ormai con prezzi nello stesso ordine di grandezza delle stampanti a matrice di media qualità.

Altre periferiche: modem e accoppiatori acustici

Se avete amici che possiedono un personal computer, potreste desiderare di mettere la vostra macchina in comunicazione con la loro, per scambiare messaggi, testi, o magari (se le due macchine sono compatibili) programmi. Chi svolge lavori professionali o di consulenza, chi lavora a casa o in un proprio studio, potrebbe desiderare di inviare i frutti del proprio lavoro direttamente al computer dei propri clienti. La cosa è possibile, almeno entro certi limiti, utilizzando una strumentazione non troppo complessa.
Gli elementi essenziali sono innanzitutto un telefono, senza il quale la comunicazione a distanza è impossibile (a meno dell'uso di linee di comunicazione dedicate, che non rientrano nelle possibilità dei privati cittadini), un'interfaccia per il calcolatore, che consente a questo di inviare all'esterno segnali adatti per il trasferimento su linea telefonica, e un dispositivo che trasformi i segnali digitali in uscita da questa interfaccia in segnali analogici che possano viaggiare sulle linee telefoniche. Una catena di dispositivi analoga è naturalmente necessaria all'altro capo della linea: un dispositivo per la trasformazione dei segnali analogici in arrivo in segnali digitali, un'interfaccia che ne permetta l'ingresso nel calcolatore.
L'interfaccia usata comunemente per le comunicazioni è un'interfaccia standard, identificata dalla sigla RS-232: si tratta di un'interfaccia che permette il trasferimento di segnali verso qualun-

Qui sopra, due Macintosh collegati mediante una semplice interfaccia Apple, che permette il collegamento in rete di un massimo di 32 apparecchiature.
Nella pagina a fronte, un piccolo calcolatore portatile (Olivetti M10) con il suo plotter (portatile) di tipo elettrostatico e, sulla sinistra, l'accoppiatore acustico su cui è posata la cornetta telefonica.

que dispositivo esterno (che potrebbe essere anche una stampante o un plotter, per esempio), codificando le informazioni elaborate dal calcolatore in una forma convenuta, comune a tutte le macchine.

Il dispositivo che trasforma i segnali digitali forniti dall'interfaccia RS-232 in segnali analogici adatti alla trasmissione su linea telefonica può essere un MODEM oppure un accoppiatore acustico. Il MODEM (sigla di modulatore/demodulatore) agisce direttamente sulla linea telefonica, generando segnali elettrici adeguati; l'accoppiatore acustico, invece, trasforma i segnali digitali in segnali sonori (in una forma analoga a quella utilizzata dai registratori a cassette per la memorizzazione) e li immette nel microtelefono come qualsiasi altro segnale acustico. Il MODEM è piú affidabile, proprio perché interviene direttamente sulla linea: è però una attrezzatura fissa, che va collegata permanentemente alla presa telefonica. In Italia, in particolare, è necessaria una omologazione da parte della SIP, che gestisce in regime di monopolio il servizio telefonico. La SIP, peraltro, mette in vendita o noleggia (esattamente come si noleggiano gli apparecchi telefonici) MODEM omologati: in caso di noleggio, il canone viene addebitato sulle normali bollette telefoniche.

L'accoppiatore acustico è meno affidabile (anche le nostre voci sulle linee telefoniche spesso diventano incomprensibili), ma è meno costoso, ed è una apparecchiatura libera, non soggetta ad omologazione, e facilmente trasportabile. Chi possiede un calcolatore portatile o trasportabile, con un accoppiatore acustico, potrebbe quindi essere autcnomo e comunicare con altri calcolatori da una qualunque cabina telefonica o da una stanza d'albergo. Esteriormente, mentre il MODEM ha semplicemente l'aspetto di una scatola senza particolarità apprezzabili, un accoppiatore acustico ha l'aspetto di un paio di coppe su cui si appoggia il microtelefono.

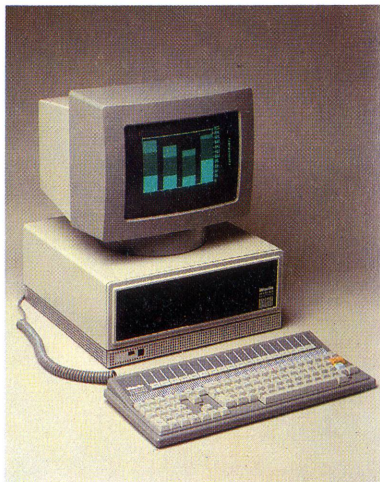
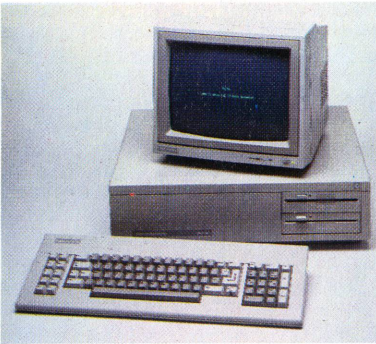
Particolarmente interessante, al di là delle comunicazioni fra amici e con datori di lavoro, è l'uso di queste apparecchiature per l'accesso a banche dati. Una banca dati è un grande archivio di

informazioni memorizzate su calcolatore: esistono circa 2000 banche dati in tutto il mondo accessibili dal pubblico, caricate su qualche centinaio di grandi calcolatori (un calcolatore può ospitare anche piú banche dati indipendenti). Esistono banche dati bibliografiche generali e specializzate, per esempio su temi come la medicina o l'energia; banche dati in cui si possono reperire informazioni sui brevetti industriali; banche dati di carattere economico e via dicendo. Per poter accedere a una banca dati è necessario stipulare un contratto con una rete di comunicazione: in Italia si deve passare attraverso l'Itapac per quanto riguarda le banche dati europee e attraverso l'Italcable per le banche dati extracontinentali. Il contratto con uno di questi enti permette solamente l'uso della rete: ci si mette in comunicazione telefonica con il nodo della rete piú vicino, poi la trasmissione avviene attraverso la rete specializzata. Per accedere alla singola banca dati, è necessario mettersi in contatto con quella banca, avere un codice d'accesso ed eventualmente pagare un canone annuale, ricevere le informazioni sul modo d'accesso e sul linguaggio da usare per interrogare la banca e ricevere informazioni. I canoni sono sempre bassi: si pagano poi (e i costi sono variabili) i tempi effettivi di collegamento con la banca, se e quando ci si collega. Il problema può essere rappresentato dai linguaggi di interrogazione, che possono variare fortemente da una banca dati all'altra: è necessario conoscere bene almeno gli elementi essenziali, per non consumare inutilmente tempo di collegamento in tentativi vani.
Per tutte queste operazioni (sia per il colloquio tra computer a livello amichevole o professionale, sia per l'accesso a banche dati) è necessario disporre di un software di comunicazione adatto: esistono programmi già confezionati per quasi tutte le macchine.

Le configurazioni

Ora che abbiamo visto, un po' piú in dettaglio, i singoli possibili componenti di

un sistema di elaborazione personale, torniamo al livello superiore, e riassumiamo quali siano gli assetti complessivi possibili, le "configurazioni" di un personal computer.
Possiamo tentare una classificazione in "categorie": è una classificazione puramente di comodo, per orientarci un po' nel mare di tipologie che si possono incontrare.

Home computer: i calcolatori domestici. Sono quelli della fascia piú bassa, prezzi contenuti, dispositivi magari ridotti all'osso: quasi sempre unità centrale e tastiera in un unico contenitore,

predisposizione per l'uso con il televisore domestico e con il registratore a cassette, "porte" per il collegamento di joystick, paddles o simili dispositivi di controllo orientati soprattutto al gioco, ma anche di un vero e proprio monitor, di unità a dischi flessibili, di una stampante. Si tratta normalmente di macchine con microprocessore a 8 bit, capacità di memoria limitate: nella configurazione di base magari ridotta a 8, 16, 32 K solamente, ma espandibile con costi aggiuntivi fino a raggiungere i 64K standard. Sono macchine perfette per la casa, per giocare, per imparare a programmare, per realizzare applicazioni non troppo impegnative. 64K, o anche meno, possono essere sufficienti per programmi anche di una certa complessità, ma sono pochi per la gestione di grandi quantità di dati. Raramente queste macchine permettono il collegamento di un disco rigido, ma in compenso hanno quasi sempre buone capacità grafiche, e permettono la gestione del colore. Sono sconsigliabili per usi professionali, ma il loro costo può essere tanto ridotto da farne oggetti acquistabili senza troppi problemi: in fondo costano anche meno di un televisore a colori, e nessuno si pone il problema "a che cosa mi serve", quando acquista un televisore a colori, ma si limita a considerare le possibilità di divertimento e di gratificazione.

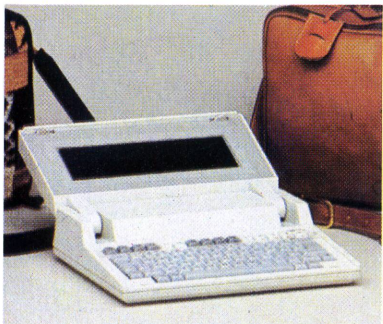
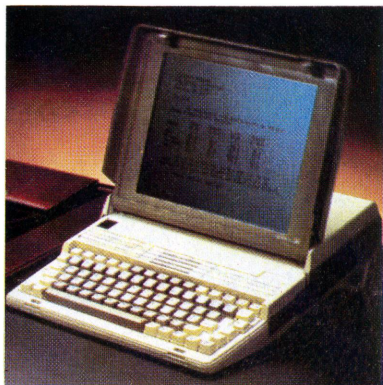
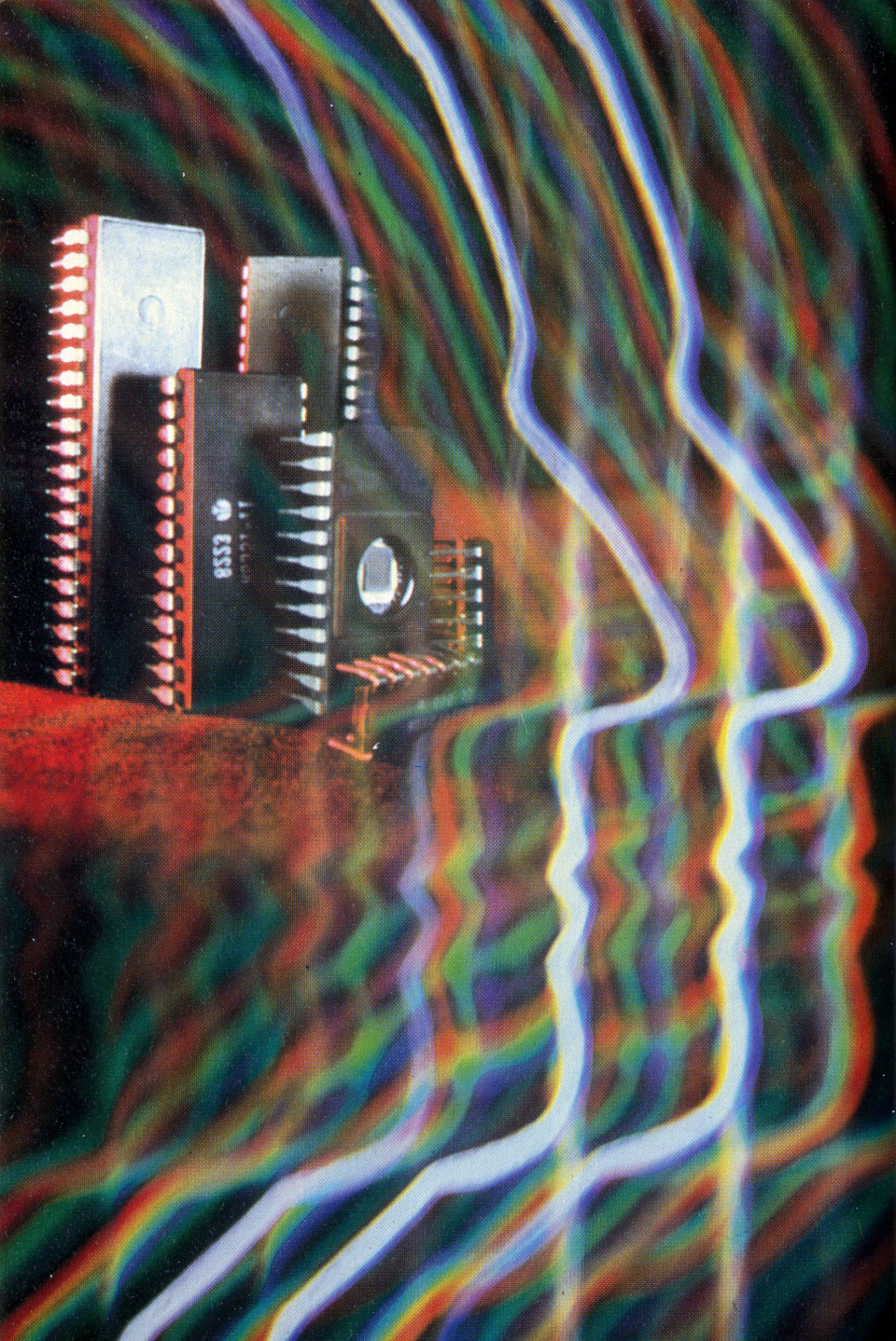
Personal computer: possiamo usare questa denominazione per indicare ge-

Alcuni esempi di possibili configurazioni di sistemi dal carattere domestico o personale-professionale. Nella pagina a fronte in alto, un sistema economico orientato soprattutto all'uso hobbistico o domestico: televisore domestico come unità video, unità centrale integrata con la tastiera (con piccoli tasti di gomma), registratore a cassette, stampantina elettrostatica. Subito qui sopra, un calcolatore medio, con tastiera incorporata all'unità centrale, monitor separato, una unità a dischi (sul lato destro) integrata e una unità a dischi esterna. Le altre tre macchine, benché esteticamente diverse tra loro, seguono lo stesso schema: tastiera e video separati, unità a dischi incorporata all'unità centrale.

nericamente tutte le macchine accessibili, per prezzo e dimensioni, al singolo, ma possiamo usarla anche per identificare quella fascia di apparecchiature che si prestano ad applicazioni di carattere professionale, con microprocessore a 8 o – meglio – a 16 bit, fornite in configurazione già completa di monitor e unità a dischi, in forma compatta o componibile. Si tratta di macchine che vanno dai 64K di memoria centrale in su, per arrivare vicino al megabyte. Non sono molto adatte per i giochi: molto spesso l'interfaccia per un joystick o altro dispositivo di controllo simile non viene neppure fornita con la macchina, ma va acquistata separatamente (e magari da costruttori diversi da quello dell'unità centrale).
Questi calcolatori permettono il collegamento di uno o piú dischi rigidi, esternamente o magari addirittura internamente, in sostituzione di un drive per dischi flessibili. Permettono il trattamento di quantità di dati relativamente grandi e l'uso di software particolarmente raffinato. Sono considerate le macchine piú adatte per i professionisti.

Trasportabili: sono calcolatori compatti, pensati appositamente per poter essere trasportati facilmente. Hanno integrati in un unico contenitore l'unità centrale, il video (in genere da 9 pollici), uno o due drive per dischetti flessibili, eventualmente un disco rigido in sostituzione di un drive per floppy disk. La tastiera può essere richiusa a modo di coperchio, e il contenitore è già fornito di una maniglia per il trasporto. Una volta chiuso, il computer assume l'aspetto di una valigia di piccole dimensioni. L'alimentazione è sempre a rete: non si può usare una di queste macchine ovunque, ma solo in un posto ove sia accessibile una presa di corrente. Il peso è sempre elevato.

Portatili: calcolatori di piccole dimensioni, fatti per trovare posto in una ventiquattrore ed essere effettivamente portati e usati ovunque. Funzionano a batterie, ma possono essere alimentati anche a rete, con apposito alimentatore. La caratteristica saliente è quasi sempre una RAM abbastanza capace e di tipo non volatile (che non perde cioè i dati quando la macchina viene spenta). I piú potenti raggiungono prestazioni paragonabili a un personal di tipo professionale, e possono essere dotati anche di un drive per dischetti flessibili da 3 pollici (il Data General/One è stato il primo portatile di questo tipo). Sono pensati

Piccoli non vuol dire necessariamente poco potenti: i calcolatori tascabili, come quello raffigurato nella pagina a fronte in basso, non sono affatto giocattoli! Possono essere programmati in BASIC, hanno qualche K byte di memoria interna, possiedono in genere qualche semplice dispositivo di memoria di massa, possono anche essere dotati di una stampantina portatile: sono eccellenti per chi deve svolgere spesso calcoli. Le altre macchine di queste due pagine sono calcolatori portatili alimentati a batteria o, alternativamente, a rete, con display a cristalli liquidi. I portatili in genere sono dotati di memoria interna non volatile, per poter conservare i dati anche dopo che è stata tolta l'alimentazione; il Data General One, subito qui sotto, è stato invece il primo ad avere una normale memoria volatile e a integrare invece una o anche piú unità a dischetti da 3 pollici come dispositivo di memoria di massa, assente in tutti i portatili precedenti.

per il professionista che viaggia molto e desidera avere sempre con sé uno strumento di lavoro comodo ed efficiente: per questo sono dotati in ROM di vari programmi, accessibili immediatamente, anche se non necessariamente integrati. La dotazione tipica comprende un programma di word processing, un tabellone eletronico, un programma per la gestione di archivi di dati e un programma di comunicazione. Queste macchine sono pensate come "secondo calcolatore": sono sempre dotate quindi di software e di hardware per le comunicazioni di dati (interfaccia RS-232 incorporata, nei modelli americani addirittura modem incorporato), perché l'utente possa facilmente scaricare tutto ciò che ha elaborato nel suo calcolatore principale, quando rientra a casa, in studio o nella sede di lavoro. Sono eccellenti per esempio per giornalisti, commessi viaggiatori che possono caricarvi gli ordini (comunicabili poi al calcolatore centrale dell'azienda via telefono), ma in generale per tutti coloro che viaggiano spesso o hanno frequentemente bisogno di trasferire i loro strumenti di elaborazione da un posto all'altro: in effetti, i piú potenti possono sostituire benissimo un personal computer da scrivania (del quale costano magari addirittura di piú, data la grande quantità di ROM e di RAM di cui possono disporre). Molte fra queste macchine sono dotate anche di una unità interna a cassette, veloce e affidabile. Ne esiste anche qualcuna che incorpora addirittura una stampante ad aghi, di piccole dimensioni o di dimensioni standard.

Dentro e fuori

Nel capitolo precedente abbiamo visto come è fatto un personal computer, ma da un punto di vista molto esterno, macroscopico. È inevitabilmente il primo livello a cui ci si accosta nel mondo del calcolatore, quello che si tocca con mano e che appare evidente nell'uso di un programma applicativo. È anche il livello su cui insistono gli annunci pubblicitari e i rivenditori: abbiamo incontrato cosí tutti gli elementi che possono aiutare a leggere criticamente una pubblicità o le parole di chi tenta di convincerci a un acquisto. Ora è venuto il momento di scendere un po' piú in profondità e di cercare di capire i meccanismi di fondo: che peraltro sono meccanismi piuttosto semplici. Tutta la raffinatezza del calcolatore elettronico deriva solo dal fatto che questi meccanismi semplici possono essere combinati in un numero elevatissimo. Anche i singoli blocchi nelle scatole di costruzioni dei bambini sono piccoli, semplici e di poche forme diverse: ma con tanti di quei blocchi si possono realizzare costruzioni notevoli: sotto Natale nelle vetrine dei negozi di giocattoli può capitare di ammirare mulini a vento, campanili con tanto di campane o navi che lasciano a bocca aperta, realizzati con perizia e pazienza con i "Lego" o materiali simili.
Per i computer la cosa non è poi tanto diversa: la memoria principale è fatta di decine o centinaia di migliaia di celle tutte uguali, e anche gli altri dispositivi

Per complesso che sia un calcolatore, i suoi componenti fondamentali sono di pochi tipi diversi e relativamente semplici.

come il microprocessore o i chip di interfaccia sono realizzati combinando centinaia, migliaia di circuiti elementari di poche forme diverse.
Alla base di tutto, come già abbiamo visto, sta il principio del commutatore, un dispositivo che può assumere due soli stati distinti, che per comodità abbiamo identificato con le cifre 0 e 1. Come sia realizzato effettivamente questo dispositivo è poco importante ai fini della struttura concettuale delle macchine: ieri si trattava di valvole termoioniche ("tubi a vuoto"), poi di relé, poi di transistor discreti, oggi di transistor e di altri dispositivi miniaturizzati, raccolti a migliaia su piccolissime piastrine di silicio, sulle quali vengono incisi con procedimenti analoghi a quello della stampa. Il silicio sarà forse sostituito in futuro da altri materiali (l'arseniuro di gallio, per esempio), che possono dare prestazioni migliori: i principi di fondo restano sempre gli stessi.
Non che la tecnologia usata sia indifferente in assoluto: i circuiti elettronici integrati sono molto piú affidabili delle valvole termoioniche (che si bruciavano spesso, bloccando tutto il funzionamento delle macchine), occupano molto meno spazio, dissipano molto meno calore (cosicché i dispositivi di raffreddamento debbono essere meno potenti e meno ingombranti), infine costano molto meno. Senza lo sviluppo dell'elettronica digitale e delle tecniche di integrazione dei circuiti, in effetti, non avremmo oggi i personal computer. Questo aspetto, tuttavia, richiederebbe un libro a sé (per chi lo desidera, ne esiste uno

XXXIV × CXIII =		
XXX	× C	= MMM
XXX	× X	= CCC
XXX	× III	= XC
IV	× C	= CD
IV	× X	= XL
IV	× III	= XII
= MMMDCCCXLII		

in questa stessa collana, interamente dedicato all'elettronica). Qui partiremo da un livello un poco superiore: quello delle funzioni di questi dispositivi.

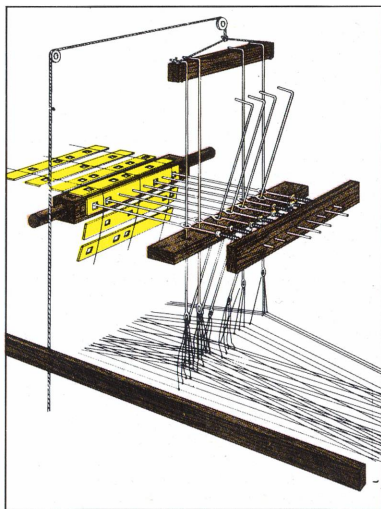
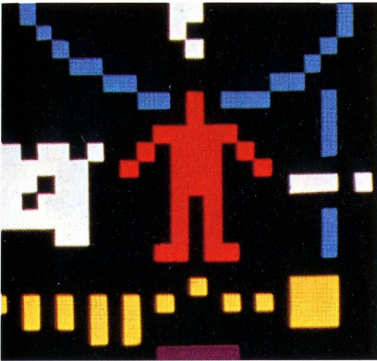
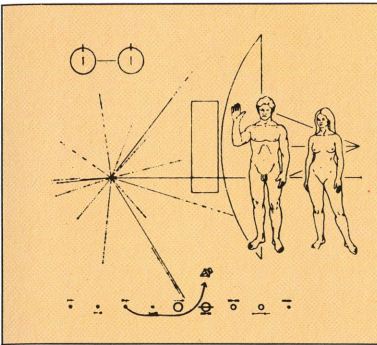
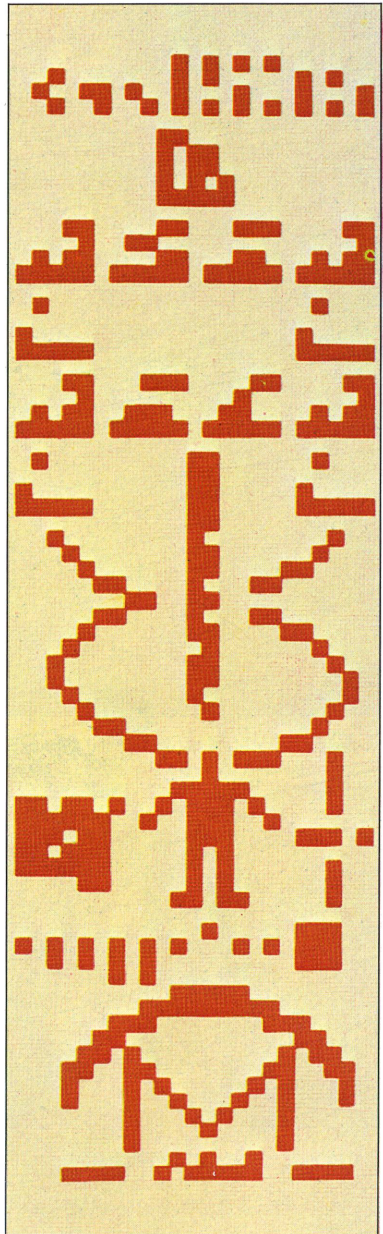
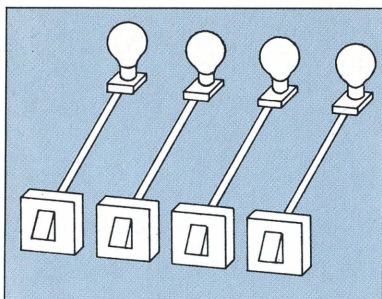
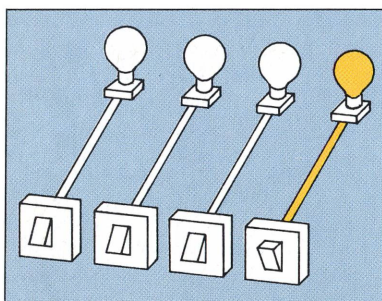
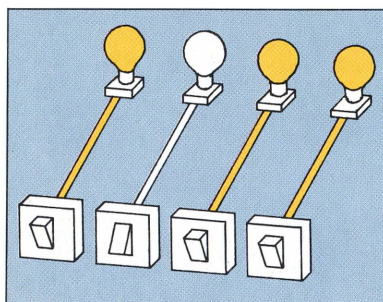
L'aritmetica in base 2

Abbiamo a disposizione due soli simboli: 0 e 1, rappresentati dal passaggio o meno di corrente (o dalla presenza di due livelli di tensione diversi, come +5 e −5 volt). La cosa piú immediata è interpretarli come due cifre: ma il nostro sistema decimale ne richiede dieci! In effetti, il sistema decimale non è che una notazione, un modo di rappresentare i numeri; ne conosciamo sicuramente almeno un altro, quello romano, basato su simboli diversi, e che si distingue dal nostro anche per non essere fondato sul principio della posizione. Anche il sistema che usiamo per i minuti e le ore è diverso: contiamo i minuti in base 60, le ore sui nostri orologi sono rappresentate in base 12.
Ci sono alcuni principi, che sottostanno a questi modi di rappresentare i numeri con cui contiamo, e che sono indipendenti dalla "base" di numerazione prescelta. Innanzitutto, il principio posizionale: le cifre sono solo dieci, nel sistema decimale, ma il significato di ogni cifra varia in funzione della posizione in cui si trova. Cosí il 3 in 23 significa "tre unità", ma in 368 significa "tre centinaia". Ogni posizione corrisponde a una potenza della "base", una potenza di 10 nel sistema decimale. Cosí, per esempio, 368 significa in effetti:

$$\begin{aligned} & 8 \times 10^0 = 8 \times 1 \\ + & 6 \times 10^1 = 6 \times 10 \\ + & 3 \times 10^2 = 3 \times 100 \end{aligned}$$

Contiamo le posizioni da destra: la prima corrisponde a 10 elevato alla zero (che è, per definizione, pari a 1); la seconda corrisponde a dieci alla prima (cioè 10); la terza a 10 alla seconda (cioè 100), la quarta a 10 alla terza (cioè 1000) e via dicendo. Il principio, in effetti, è indipendente dalla base scelta: possiamo scegliere come base il 2 e costruire un sistema di numerazione posizionale del tutto analogo, ma in cui ci servono solo due cifre, lo 0 e l'1. Ciascuna posizione, in questo sistema, corrisponde a una potenza di 2: la prima posizione a 2 alla zero (cioè 1), la seconda a 2 alla prima (cioè 2), la terza a 2 alla seconda (cioè 4), e via di questo passo. 1010 in notazione binaria significa:

$$\begin{aligned} 0 \times 2^0 &= & 0\,+ \\ 1 \times 2^1 &= 1 \times 2 = & 2\,+ \\ 0 \times 2^2 &= 0 \times 4 = & 0\,+ \\ 1 \times 2^3 &= 1 \times 8 = & 8 = \\ & & \overline{10} \end{aligned}$$

Il numero binario 1010 corrisponde, dunque, al numero decimale 10. La ta-

Il principio posizionale stabilisce che a ciascuna posizione, nella scrittura di un numero, corrisponde un valore ben preciso: nel sistema decimale, la prima cifra a destra è sempre quella delle unità, la seconda da destra è sempre quella delle decine, ecc. Il sistema di numerazione romano (a fronte) non aveva natura posizionale. Il sistema binario segue il principio posizionale, ma si basa sulle potenze di due, anziché su quelle di 10. Un antecedente storico al suo uso nei calcolatori si trova nel telaio Jacquard, del Settecento, in cui il movimento degli aghi era controllato da schede perforate: l'ago era messo in movimento se era presente una perforazione, altrimenti rimaneva bloccato. Se si pensa alla perforazione come 1 e alla sua assenza come 0, l'attività del telaio appare codificata da numeri binari. Sul sistema binario posizionale si basano anche i messaggi inviati nel cosmo con la sonda Pioneer 10 (sopra) e dal radiotelescopio di Arecibo (a fianco).

Una lampadina può rappresentare una cifra binaria: 0 se spenta, 1 se accesa. Con 4 lampadine si rappresentano 16 cifre.

La lampadina che rappresenta le unità è accesa, le altre sono spente: 0001. È la forma binaria del numero 1.

Questa configurazione ci dà 1011, ovvero 1×8+0×4+1×2+1×1: è la rappresentazione binaria del numero 11 decimale.

I PRIMI 20 NUMERI NEI QUATTRO SISTEMI FONDAMENTALI DI NUMERAZIONE

decimale	binario	ottale	esadec.
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6.
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

bella qui sopra riportata mostra la forma binaria corrispondente ai primi venti numeri decimali.
L'aritmetica binaria è del tutto analoga a quella decimale: è, anzi, piú semplice, perché gli unici casi elementari di cui si debba tener conto sono quelli che si riferiscono alle uniche due cifre utilizzate. Cosí, per esempio, nell'addizione si possono dare solamente quattro casi elementari:

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 0$
$1 + 1 = 10$

che si riducono poi a tre, dato che l'addizione gode della proprietà commutativa e quindi $0 + 1 = 1 + 0$. Le somme di numeri a piú cifre sfruttano questo meccanismo, con il "riporto" nel caso di

Addizione

Addizioni fondamentali:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 con riporto di 1

Esempio:

```
100100 +
  1101 =
------
110001
```

Sottrazione

Differenze fondamentali:

0 - 0 = 0
1 - 0 = 1
1 - 1 = 0
0 - 1 = 1 con prestito di 1 dalla cifra precedente

Esempio:

```
10001 —
 1101 =
-----
00100
```

(10001 binario è 17 decimale; 1101 binario è 13 decimale; 17 - 13 = 4, che in binario è 100).

Moltiplicazione

Come nel sistema decimale: in pratica si ha solo una serie di addizioni del moltiplicando incolonnato opportunamente, perché quando si moltiplica per 0 si ha 0 (si salta un posto) e moltiplicando per 1 si ottiene ancora il numero di partenza.

Esempio:

```
   10101 ×
    1001 =
--------
   10101
10101---
--------
10111101
```

Divisione

Analoga a quella decimale, ma piú semplice perché esistono solo i valori 0 e 1.

Esempio:

```
11110 : 10 = 1111
10
--
 11
 10
 --
  11
  10
  --
   10
    =
```

1 + 1. Qui sopra si possono vedere i casi elementari delle quattro operazioni, e alcuni esempi.

Base 8 e base 16

Il sistema binario è concettualmente semplice, ma noioso da usare in pratica: per comodità, si usano spesso in informatica i sistemi di numerazione in base 8 e in base 16, che risultano meno farraginosi, piú agili da maneggiare perché presentano maggiori somiglianze con il sistema decimale, e tuttavia cosí vicini al sistema binario (8 e 16 sono potenze di 2), da rendere facilissimo il passaggio da un sistema all'altro.
Il sistema a base 8, chiamato anche ottale, fa uso delle cifre da 0 a 7: per rappresentare i numeri dallo 0 al 7, nel sistema binario sono necessari tre bit (tre cifre binarie), e il passaggio dal sistema binario all'ottale si attua semplicemente raggruppando le cifre binarie a tre a tre (partendo dalla destra) e sostituisce ciascun gruppo di tre con la corrispondenza cifra ottale. Il passaggio dal sistema ottale al binario si effettua esattamente al contrario: a ogni cifra ottale si sostituisce il corrispondente gruppo di tre cifre binarie.
Nel sistema a base 16 (esadecimale o hex, secondo l'abbreviazione dell'inglese *hexadecimal*) sono necessarie 16 cifre distinte: si usano le dieci del sistema decimale e le prime sei lettere dell'alfabeto (A = 11, B = 12, C = 13, D = 14, E = 15, F = 16). Anche per questo sistema la conversione con il sistema binario è molto semplice: a ogni cifra esadecimale corrispondono quattro cifre binarie. Per passare dal binario all'esadecimale si raggruppano le cifre binarie a quattro a quattro a partire da destra e

si sostituisce ciascun gruppo di quattro con la corrispondente cifra esadecimale. Per convertire da esadecimale a binario si opera esattamente all'inverso.
Le operazioni aritmetiche nel sistema ottale ed esadecimale non nascondono particolari sorprese: per l'ottale, poi, la maggior parte dei calcoli si presenta proprio come nel sistema decimale. I meccanismi dell'aritmetica ottale ed esadecimale sono presentati nelle figure di queste pagine, con qualche esempio.
In pratica, poiché è sempre possibile convertire le rappresentazioni da un sistema all'altro, si usa il sistema piú comodo e si procede alla fine alla conversione nel sistema che serve. Il sistema binario è la forma di rappresentazione compresa effettivamente dalla macchina, ma non è difficile costruire dispositivi di decodifica che permettano di immettere dati numerici in una forma piuttosto che nell'altra, convertendoli automaticamente nel sistema necessario per la macchina.
Non staremo a entrare oltre nei particolari: l'importante è capire che con il sistema binario si possono fare esattamente le stesse cose che si possono fare con il sistema decimale.

Cifre e lettere

Tutti i numeri possono essere rappresentati, grazie al sistema binario, con due sole cifre. È già un buon passo avanti. E per i caratteri alfabetici? Basta identificarli con i numeri. Probabilmente abbiamo provato tutti a divertirci con qualche forma elementare di crittografia: si fa corrispondere un numero alla lettera A, un altro alla B, e via dicendo. Lo si fa normalmente con i numeri decimali, ma abbiamo visto che i numeri decimali possono essere convertiti in numeri binari: quindi lo stesso vale per i caratteri. In effetti, la cosa piu importante è stabilire un "codice" che metta in corrispondenza caratteri e numeri, e usare sempre quello.
Il principio è analogo a quello del telegrafo: punti e linee possono essere assimilati benissimo agli 0 e 1 del codice binario adatto per i calcolatori. Esistono alcuni codici standard, su cui si è stabilito un accordo internazionale: quello usato comunemente per i personal (e piú in generale per tutti i computer) è il codice ASCII (che si pronuncia "aschi", ed è una sigla per *American Standard Code for Information Interchange*, "codice standard americano per lo scambio di informazioni"). Il codice ASCII usa parole di 8 bit per la codifica dei caratteri; con 8 bit si possono rappresentare in tutto 2^8, cioè 256 caratteri. In effetti il codice fissa univocamente 128 caratteri, per i quali sono sufficienti 7 bit. L'ottavo è libero, e può servire come bit di controllo nel campo delle trasmissioni, può essere lasciato fisso e inutilizzato, oppure può fungere da "commutatore" fra due insiemi diversi di caratteri (il secondo dei quali non è standardizzato). In effetti su molti personal computer il codice ASCII viene esteso, sfruttando tutti i 256 codici possibili: i primi 128 restano quelli standard, i successivi codificano simboli speciali, per esempio forme nazionali come le lettere accentate o quelle con l'Umlaut tedesche, e in genere caratteri grafici. In questa seconda serie di caratteri non esiste alcun tipo di standardizzazione: ogni macchina presenta caratteristiche proprie.

Le combinazioni di 0 e 1

Fin qui, nulla di straordinario: abbiamo visto come sia possibile rappresentare con due sole cifre, 0 e 1, tutti i numeri e i caratteri che usiamo nella comunicazione quotidiana. Nulla di difficile: tutt'al piú un po' noioso. Rappresentare, comunque, non è ancora elaborare: ed è questo invece che ci interessa.
Partiamo solo con cifre binarie: anche in uscita dovremo avere cifre binarie. I meccanismi interni di un calcolatore sono giocati tutti su questo. Prendiamo un solo bit, uno 0 o un 1. Quali elaborazioni sono possibili? Solo due. Una è poco significativa: lasciare il bit identico. L'altra elaborazione possibile consiste nel

Tabella dei 128 codici ASCII: a ciascun simbolo è associato un numero, qui dato in forma decimale.

CODICE ASCII

DECIMALE	ASCII	DECIMALE	ASCII	DECIMALE	ASCII
0	NUL	43	+	86	V
1	SOH	44	,	87	W
2	STX	45	-	88	X
3	ETX	46	.	89	Y
4	EOT	47	/	90	Z
5	ENQ	48	0	91	[
6	ACK	49	1	92	\
7	BEL	50	2	93	]
8	BS	51	3	94	∧
9	HT	52	4	95	—
10	LF	53	5	96	`
11	VT	54	6	97	a
12	FF	55	7	98	b
13	CR	56	8	99	c
14	SO	57	9	100	d
15	SI	58	:	101	e
16	DLE	59	;	102	f
17	DC1	60	<	103	g
18	DC2	61	=	104	h
19	DC3	62	>	105	i
20	DC4	63	?	106	j
21	NAK	64	@	107	k
22	SYN	65	A	108	l
23	ETB	66	B	109	m
24	CAN	67	C	110	n
25	EM	68	D	111	o
26	SUB	69	E	112	p
27	ESC	70	F	113	q
28	FS	71	G	114	r
29	GS	72	H	115	s
30	RS	73	I	116	t
31	US	74	J	117	u
32	spazio	75	K	118	v
33	!	76	L	119	w
34	“	77	M	120	x
35	#	78	N	121	y
36	$	79	O	122	z
37	%	80	P	123	{
38	&	81	Q	124	|
39	'	82	R	125	}
40	(	83	S	126	~
41	)	84	T	127	DEL
42	*	85	U		

Le tavole di verità

p AND q

p	q	p AND q
1	1	1
1	0	0
0	1	0
0	0	0

p OR q

p	q	p OR q
1	1	1
1	0	1
0	1	1
0	0	0

NOT p

p	NOT p
1	0
0	1

trasformare una cifra nell'altra: 0 in 1 e 1 in 0. Non ci sono altre possibilità.
Ci serve un dispositivo "invertitore", dunque, per realizzare la piú semplice di tutte le elaborazioni possibili.
Complichiamoci un po' la vita. Due segnali in ingresso, ciascuno dei quali può essere 0 e 1: le possibili combinazioni in ingresso sono solo 4. Se chiamiamo A e B le due linee di ingresso, le possibilità sono:

linea	A	B
	0	0
	0	1
	1	0
	1	1

Non ne esistono altre. Ora immaginiamo che esista un dispositivo molto semplice di elaborazione, che ha come ingressi le nostre due linee A e B. Possiamo immaginare questo dispositivo come una scatola nera: vediamo che cosa entra e che cosa esce, non sappiamo che cosa c'è dentro. La scatola nera ha una sola linea di uscita: prende quindi due segnali in ingresso, e fornisce uno 0 o un 1 in uscita.
Quanti dispositivi di questo genere possiamo costruire? Tante quante sono le possibili combinazioni di uscite in funzione delle combinazioni di entrate. Per ciascuna delle righe della tabellina precedente (che ci dava le possibili combinazioni di bit in ingresso) esistono due possibilità di uscita (come sempre, 0 o 1): in totale abbiamo $2 \times 2 \times 2 \times 2 = 16$ possibilità.
Esiste un modo semplice e intuitivo per raffigurare queste possibilità, che prende il nome di *tavola di verità* (vedremo piú avanti il perché di questo nome). Una tavola di verità riporta su ciascuna riga una combinazione di entrate e di uscite. Facciamo un esempio:

in ingresso A	B	in uscita
0	0	1
0	1	0
1	0	0
1	1	1

Questa tavola di verità descrive il comportamento di un dispositivo di elaborazione che accetta in ingresso due segnali e produce un segnale in uscita, in funzione della natura dei segnali in ingresso: nel nostro caso, un dispositivo che ci dà in uscita il valore 1 se e solo se i segnali in ingresso sono uguali, e ci dà il valore 0 in caso contrario.
Esistono 16 possibili tavole di verità distinte per dispositivi di questo genere (con due entrate e un'uscita), ma si dà il caso che tutte queste tavole di verità non siano indipendenti: se ne possono individuare alcune che descrivono il comportamento di dispositivi particolari, che opportunamente combinati permettono di dare (a parità di segnali in ingresso) le stesse uscite di altri dispositivi descritti dalle altre tavole di verità.
Facciamo un passo indietro: 0 e 1 possono servire a rappresentare numeri e caratteri di tutti i tipi. Ma possiamo interpretarli anche come abbreviazioni di "falso" e "vero", rispettivamente; i due "valori di verità" che una frase può assumere. Siamo nel campo della logica, lo studio dei processi di ragionamento e delle loro caratteristiche formali.

p XOR q

p	q	P XOR q
1	1	0
1	0	1
0	1	1
0	0	0

p NOR q

p	q	P NOR q
1	1	0
1	0	0
0	1	0
0	0	1

p NAND q

p	q	p NAND q
1	1	0
1	0	1
0	1	1
0	0	1

Una frase (un enunciato, come si dice formalmente per indicare una frase dichiarativa) può assumere solo due valori di verità: o è vera o è falsa (o almeno cosí ci dice la teoria classica, e non abbiamo ragione qui per metterla in dubbio). Ma, senza andare ad analizzarne la struttura interna, che cosa si può dire (restando su un piano puramente formale) del valore di verità di una frase composta da piú frasi collegate fra loro da una congiunzione come "e" oppure "o", "se" e simili?
Che valore può avere una frase "p e q", dove p e q sono frasi elementari? Per esempio, che cosa si può dire di "Giovanni vive a Parigi e Maria vive a Londra"? Che la frase "p e q" è vera se e solo se sono vere sia p sia q, cioè che "Giovanni vive a Parigi e Maria vive a Londra" è una frase vera se e solo se "Giovanni vive a Parigi" è una frase vera e "Maria vive a Londra" è una frase vera. È da notare che questa affermazione non ha nulla a che vedere con il significato specifico di ciascuna frase, ma riguarda solamente il modo in cui le frasi sono collegate fra loro, cioè precisa il ruolo che svolge la congiunzione "e". Questo ruolo è quello di creare, a partire da due frasi, una terza frase composta che è vera se e solo se ambedue le frasi di partenza sono vere.
Anche il comportamento della "e" in questa analisi logica può essere sintetizzato con una "tavola di verità", in cui vengono riportate le possibili combinazioni di condizioni di verità degli enunciati di partenza e i valori di verità corrispondenti per l'enunciato composto. Possiamo indicare il vero con V e il falso con F:

p	q	peq
V	V	V
V	F	F
F	V	F
F	F	F

La tavola si legge per righe: se p e q sono vere, "p e q" è vera; se p è vera e q è falsa, "p e q" è falsa, e via dicendo.
Al posto di V e F possiamo mettere qualunque altro simbolo, in particolare 1 (per V) e 0 (per F): otteniamo esattamente una delle tavole di verità per i nostri circuiti elementari.
La congiunzione "e" corrisponde all'inglese *and*: e il tipo di circuito elementare che ha come tavola di verità quella appena vista per la congiunzione "e" viene indicato comunemente con il nome AND. Questa breve digressione spiega anche perché si attribuisca il nome di "tavola di verità" alla tabella che fornisce il rapporto fra ingressi e uscite di un circuito digitale.
La stessa analisi che abbiamo fatto per la "e" si può fare per la "o": una congiunzione che crea frasi composte il cui valore di verità è il vero se e solo se almeno una delle frasi di partenza è vera. Cosí "Vado al cinema o vado a teatro" è una frase vera se almeno una delle frasi "Vado al cinema", "Vado a teatro" è vera: è vera anche se si verificano tutte e due le possibilità. La tavola di verità corrispondente dà in uscita il valore 1 in tutti i casi, fuorché nel caso in cui ambedue gli ingressi siano 0.
Il circuito corrispondente prende il nome di OR (corrispondente inglese del nostro "o").
La "o" in questo caso ha valore inclusi-

vo: corrisponde al *vel... vel* dei latini. Molto spesso, però, usiamo la "o" in senso esclusivo, intendendo che una sola delle due alternative può essere vera: i latini avevano la congiunzione *aut... aut* per questi casi. La tavola di verità relativa dà valore 1 in uscita solo quando i valori in ingresso sono diversi, mentre dà valore 0 quando i valori in ingresso sono uguali (si ha il vero solo quando una frase è vera e l'altra e falsa). Anche questa tavola di verità descrive un circuito particolare, che viene indicato con il nome XOR (contrazione di *exclusive or*, cioè "o esclusivo").

Per concludere questo breve confronto fra circuiti e logica degli enunciati, dobbiamo ancora prendere in considerazione il semplice circuito che prendeva in ingresso un bit e lo trasformava nel bit opposto (dava 1 in uscita per 0 in ingresso, e viceversa). Questo circuito ha un semplice corrispondente nella lingua ordinaria: la preposizione "non" o, per maggiore semplicità, una locuzione negativa come "non è vero che". Premettendo questa locuzione a una frase qualunque, si ottiene una nuova frase il cui valore di verità è opposto a quello della frase di partenza. Se "Nel 1984 Craxi era presidente della repubblica italiana" ha valore di verità falso, "Non è vero che nel 1984 Craxi era presidente della repubblica italiana" ha valore di verità vero. E viceversa.

Il corrispondente inglese della negazione "non" è *not*: e NOT è anche il nome che si attribuisce comunemente al semplice circuito che inverte fra loro i valori 0 e 1 (il circuito viene chiamato spesso anche "invertitore").

Algebra di Boole

La divagazione non è stata inutile: abbiamo visto come esista una curiosa corrispondenza fra logica degli enunciati e analisi dei circuiti digitali elementari. Questa corrispondenza può essere sfruttata ampiamente: quello che abbiamo scoperto è che possiamo trasferire da un campo all'altro le nostre scoperte, perché i due campi hanno la stessa struttura generale. In termini matematici, si tratta di due diverse "incarnazioni" di una medesima struttura algebrica: un insieme di elementi su cui sono definite alcune operazioni. Una struttura algebrica è indipendente dalla natura degli elementi dell'insieme: è una struttura astratta. Il vantaggio di studiare strutture algebriche astratte sta nel fatto che si possono dimostrare teoremi e raggiungere risultati di validità generale: per qualunque insieme che in concreto presenti realizzata quella struttura astratta quei risultati, quei teoremi generali varranno sicuramente.

È sufficiente sapere che la Dinamo Bucarest è una squadra di calcio, per sapere anche che schiera in campo 11 uomini, uno dei quali è sicuramente un portiere; che ha un allenatore, e altre cose del genere. Se si va ad assistere a una partita di calcio, non verrà sicuramente in mente di chiedere se la squadra avversaria ha un portiere: non sarebbe una squadra di calcio se non l'avesse, come non sarebbe una squadra di calcio se in campo scendessero solo cinque giocatori (sarebbe probabilmente una squadra di pallacanestro). Cosí succede per le strutture algebriche: sapere che un insieme di oggetti costituisce un gruppo algebrico, per esempio, significa sapere automaticamente che sugli elementi di quell'insieme di oggetti è definita un'operazione che gode della proprietà associativa, per la quale ogni oggetto ha un suo inverso, e che esiste un elemento identità (che combinato tramite l'operazione prescelta con qualunque altro elemento lo lascia inalterato, come fa lo 0 nella somma definita sui numeri naturali). La natura degli elementi che costituiscono l'insieme non è importante, o almeno non lo è in questo contesto, come il nome dei giocatori non è importante per avere una squadra di calcio.

Il tipo di struttura algebrica che soggiace sia alle combinazioni di enunciati mediante congiunzioni come "e" e "o", sia alle combinazioni di segnali digitali mediante circuiti con due ingressi e un'uscita, è l'*algebra di Boole*, dal nome di George Boole, matematico inglese che per primo l'ha studiata nell'Otto-

cento (molto prima che trovasse applicazioni nel campo dei circuiti elettronici e dei calcolatori).
Lo studio dell'algebra di Boole come struttura astratta ci dice qualche cosa di molto importante anche per i nostri circuiti elementari: se le combinazioni possibili sono in tutto 16, ciò non significa che abbiamo bisogno di 16 tipi diversi di circuiti per realizzarle tutte. In effetti, sono sufficienti due soli tipi di circuiti: l'AND e il NOT, oppure l'OR e il NOT; o piú semplicemente ancora, è sufficiente un unico circuito, che svolga le funzioni di AND e NOT insieme (circuito NAND) o di OR e NOT insieme (circuito NOR). Qualunque altra combinazione si può ottenere disponendo opportunamente piú dispositivi di questo tipo in un unico circuito.
Se anziché due gli ingressi debbono essere tre o piú, il problema non cambia, perché c'è sempre un modo per elaborare i primi due, combinare il risultato con il terzo, e via di seguito: gli stessi dispositivi elementari servono anche per realizzare questi circuiti piú complessi. Se poi gli ingressi sono piú numerosi e debbono subire piú elaborazioni, per dare piú di una uscita, si applica sempre lo stesso principio, disponendo piú gruppi di circuiti elementari in parallelo.
Supponiamo di avere due numeri binari (risultato magari di qualche elaborazione precedente) di otto cifre ciascuno, cioè lunghi un byte, e di avere otto circuiti del tipo XOR (OR esclusivo). Mandiamo in ingresso a ciascun circuito XOR le due cifre corrispondenti dei due numeri: nel primo XOR la prima cifra di ciascun numero, nel secondo la seconda cifra e cosí via di questo passo. Il risultato possiamo pensare di inviarlo in otto celle contigue di memoria.
Che cosa abbiamo ottenuto? Il circuito XOR dà in uscita uno 0 se i suoi ingressi sono uguali; 1 in caso contrario. Allora dal contenuto delle otto celle di memoria adesso possiamo stabilire se i due numeri binari in ingresso erano uguali o diversi!
Finché il risultato è depositato nelle otto celle di memoria non ci serve a molto: ci serve anche qualche tipo di dispositivo che controlli se tutte le cifre del byte risultato sono 0, oppure se ce n'è qualcuna uguale a 1. Il nostro byte potrebbe essere portato in un punto particolare del sistema di elaborazione, all'interno di un dispositivo particolare di memoria in grado di contenere solo un byte: è quello che si definisce un *registro* dell'unità di elaborazione. Potremmo pensare di far scorrere verso sinistra tutto il numero, inserendo a destra nella cella che ogni volta si libera uno zero. A sinistra, a ogni passaggio, il registro "straripa": un bit non ci sta piú e deve essere fatto uscire. Lo possiamo portare in una cella apposita: qui possiamo controllare se si tratta di uno 0 o di un 1. Ripetendo tutto il procedimento otto volte, fino a esaurimento dei bit del byte di partenza, possiamo controllare se tutte le cifre erano 0.
L'esempio non è puramente illustrativo: il *logical shift* o *scorrimento logico* è una fra le operazioni fondamentali di tutti i microprocessori. Se ricordate, abbiamo detto che nel codice ASCII i singoli caratteri sono codificati come successioni di otto bit: tutto questo procedimento, quindi, servirebbe anche per controllare se due lettere sono identiche. Tutto il funzionamento del calcolatore è, alla base, di questo genere: le singole operazioni possibili sono molto elementari, e anche per controllare se due lettere sono identiche è necessaria una serie abbastanza lunga di operazioni elementari. Il vantaggio del calcolatore è la velocità a cui esegue queste operazioni: può eseguirne migliaia al secondo, molte piú di quelle che saremmo in grado di eseguire noi con carta e matita! Cosí la macchina può andare a controllare tutto un testo alla ricerca di una data parola, in men che non si dica: ma quel che fa realmente è andare a controllare lettera per lettera, cioè in sostanza bit per bit.
Il numero binario 01100001 rappresenta il numero 97 decimale (61 in esadecimale) e nel codice ASCII corrisponde alla lettera "a" minuscola. Che cosa succede se prendiamo il numero 11011111 ed effettuiamo l'AND di questi due bite, bit per bit?

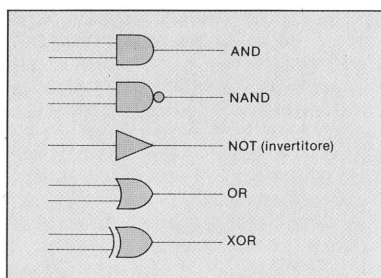
```
     01100001
AND  11011111
-------------
     01000001
```

Il risultato, 01000001 binario, è pari a 65 decimale (41 esadecimale) e, nel codice ASCII, corrisponde alla lettera "A" (maiuscola). Si può provare a eseguire la stessa operazione con i codici delle altre lettere minuscole: il risultato è sempre lo stesso, la trasformazione in lettera maiuscola. Semplice, no? Ma anche abbastanza lontano dal nostro modo di lavorare quotidiano! Questo, in effetti, è un buon esempio del modo particolare in cui debbono essere rivisitati tutti i nostri modi di operare, per adattarli ai meccanismi fondamentali di funzionamento del calcolatore.
Alla stessa stregua, come si può trasformare una lettera maiuscola nella corrispondente minuscola? Bisogna far intervenire la funzione XOR (OR esclusivo). Riprendiamo il codice ASCII per la "A" maiuscola, 01000001, e consideriamo il codice 00100000:

```
     01000001
XOR  00100000
-------------
     01100001
```

Il risultato è proprio il codice ASCII per la "a" minuscola.

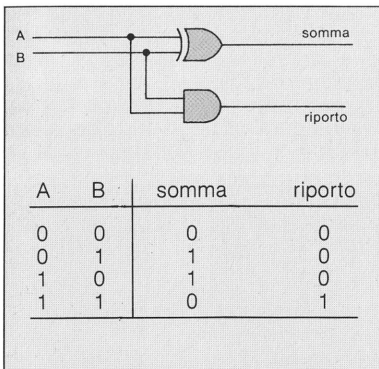
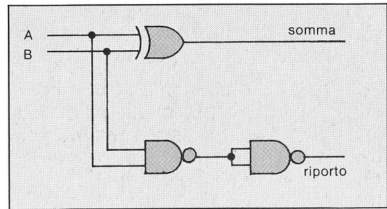
Circuiti per sommare

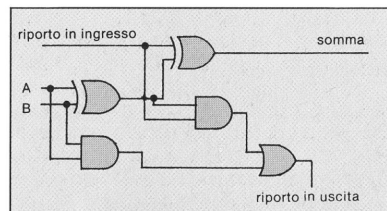
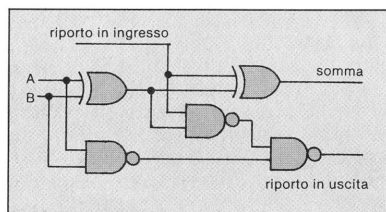
In teoria, si potrebbe costruire tutta l'unità di elaborazione di un calcolatore sfruttando unicamente circuiti di tipo NAND, ma nella pratica il numero dei circuiti elementari utilizzati è un po' piú ampio: AND, OR, NOT, XOR e NOR trovano tutti qualche applicazione. La scelta può variare ampiamente da costruttore a costruttore, e siccome, per esempio, ci vogliono piú circuiti NAND per realizzare un circuito equivalente allo XOR, da una macchina all'altra l'efficienza di una medesima operazione può risultare leggermente diversa. Il nostro interesse per i circuiti elementari, comunque, in questo contesto, non è

I calcolatori elettronici sono costruiti con un numero grandissimo di pochi tipi di componenti elementari (circuiti logici o porte logiche). Nella figura sono presentati i simboli usati dai progettisti per le cinque porte logiche fondamentali: il NOT ("non") o invertitore ha un unico ingresso e dà come uscita il valore opposto; AND ("e") dà in uscita 1 quando riceve in ingresso due segnali 1; OR ("o") dà in uscita 1 quando almeno uno dei suoi due ingressi riceve un segnale 1; XOR ("o esclusivo") dà in uscita 1 quando uno solo dei suoi ingressi riceve un segnale 1; NAND ("not and") dà sempre in uscita 1, fuorché quando ambedue i suoi ingressi ricevono un segnale 1.

cosí raffinato, quindi possiamo permetterci di far uso dei circuiti piú comodi per questa prima esplorazione.
Non abbiamo ancora visto una elaborazione interpretabile in senso strettamente numerico: è venuto il momento di cimentarci con la piú semplice fra le operazioni aritmetiche, la somma. Cominciamo con il considerare la somma fra due soli bit, che arriveranno al nostro circuito tramite gli ingressi A e B. In un caso (ambedue gli ingressi 1), la somma dei due bit deve essere rappresentata ancora da due bit: c'è un "riporto", di cui dobbiamo tener conto nella costruzione. Ecco la tavola di verità modificata in conseguenza:

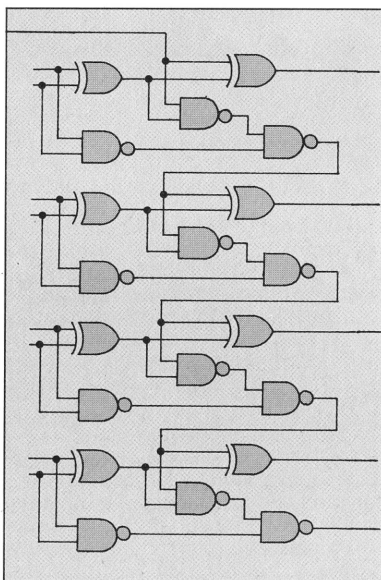
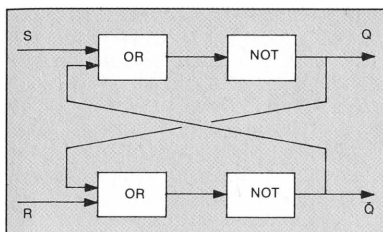
ingressi		uscita	riporto
A	B		
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Ragioniamo separatamente sull'uscita e sul riporto. L'uscita deve essere 1 quando i segnali in ingresso sono diversi; 0 quando i segnali sono uguali. Questo è esattamente il comportamento del circuito XOR. Il riporto (*carry*, in inglese) deve invece essere 0 in tutti i casi, fuorché quando ambedue i segnali in ingresso sono 1. Anche qui abbiamo un circuito elementare che fa proprio al caso nostro: questo è esattamente il comportamento del circuito AND.
Quindi dobbiamo far arrivare i nostri due segnali in ingresso a due circuiti elementari diversi, uno XOR e un AND, che debbono lavorare in parallelo: il primo dei due ci darà il bit di somma, il secondo ci darà il bit di riporto. Ecco come si può rappresentare graficamente il circuito complessivo risultante:

A	B	somma	riporto
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Questo circuito viene chiamato *half-adder*, cioè "semisommatore" e non semplicemente "sommatore" perché in realtà non tiene conto del fatto che in una somma effettiva, in tutte le posizioni tranne la prima, è possibile avere un riporto: quindi bisogna tener conto anche di quest'ulteriore elemento. Prima di vedere come si possa realizzare un *full adder*, un sommatore completo, è interessante però vedere come il circuito semisommatore possa essere realizzato con circuiti elementari diversi. Ecco come si potrebbe ottenere lo stesso risultato usando solamente circuiti XOR e NAND:

Il circuito sarebbe piú complesso usando solo NAND. In questo caso è facile stabilire quale delle soluzioni sia piú economica, ma i progettisti, alle prese con schemi complicati, con un gran numero di circuiti elementari, spesso si debbono porre il problema: la soluzione che ho trovato è realmente la piú semplice, non esiste un altro modo di realizzare la stessa funzione, che faccia uso di un minor numero di circuiti elementari? Sono stati sviluppati, in effetti, dei metodi per stabilire se un circuito è il piú semplice possibile, o se esistono modi piú efficienti per ottenere lo stesso risultato: uno di questi sistemi sfrutta la corrispondenza fra logica degli enunciati e circuiti digitali. Torniamo al nostro sommatore: vediamo ora come si possa costruire un sommatore completo, tenendo conto anche del fatto che si può avere un riporto da una somma precedente. Gli ingressi a ciascun circuito elementare sono sempre solo due, quindi dobbiamo frammentare il problema in piú passi distinti: possiamo sommare prima i due segnali in ingresso, poi sommare il risultato con il bit di riporto. Per il nuovo riporto che si genera, dobbiamo tener conto dei due segnali in ingresso, poi del risultato della loro somma con il riporto in ingresso. Ecco come si può realizzare il sommatore completo, con circuiti XOR, AND e OR:

Abbiamo, in effetti, due circuiti semisommatori combinati, con un OR finale che permette di stabilire se in uno dei due si è generato un bit di riporto (è sufficiente un OR dal momento che non si può verificare il caso di due bit di riporto; solo uno dei due semisommatori può dare un 1 di riporto – provare per credere). Per proseguire nel confronto, ecco invece come lo stesso circuito può essere realizzato con circuiti XOR e NAND: è un poco piú difficile da seguire, ma va notato come il numero dei circuiti elementari ora sia identico, nelle due soluzioni.

L'ultimo passo, per ottenere un circuito sommatore reale, sta nel disporre un gruppo di sommatori in parallelo, con il riporto di ciascuno collegato al successivo: ce ne servono otto, se lavoriamo con un calcolatore che abbia parole di otto bit. A questo punto non c'è nulla di nuovo; ma soltanto ripetizione dello stesso schema otto volte (che possono diventare 16 o 32 se vogliamo lavorare con parole di questa lunghezza). Se ci fa comodo, possiamo predisporre piú sommatori da 8 bit in parallelo, ed elaborare piú numeri contemporaneamente. Un sommatore a 4 bit è schematizzato nella figura qui a fianco: il sommatore a 8 bit non è altro che il doppio.
Va notato che può presentarsi un riporto anche nell'ultima posizione (corrispondente alla cifra piú alta, la piú significativa): in questo caso si ha uno "straripamento" (*overflow* in inglese). In questo caso il risultato non può esser preciso, per la limitazione che ci siamo imposti, cioè contenere i numeri in un byte. In qualche modo la cosa deve essere segnalata, e per fare questo possiamo mandare il valore del riporto in un registro apposito, il registro di overflow: possiamo poi stabilire che, prima di operare ulteriormente sul risultato dell'operazione, il calcolatore vada a controllare il registro di overflow e, se vi trova un 1, lo segnali; se invece trova uno 0 è autorizzato a procedere.

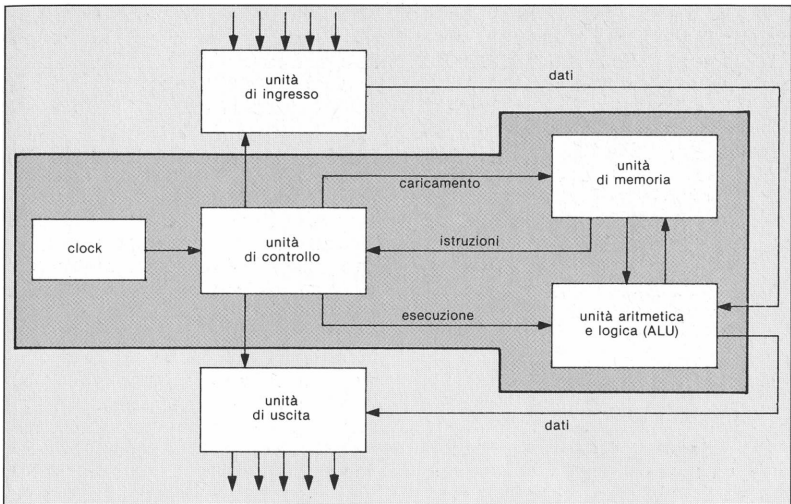
Flip-flop: la memoria

Fin qui abbiamo parlato di circuiti che elaborano informazioni, e abbiamo visto qualche caso particolarmente semplice, ma significativo. Questi circuiti, tuttavia, non "ricordano" gli eventi del loro passato, mentre a noi serve anche qualche dispositivo che consenta di archiviare le informazioni binarie. Un circuito che possa fungere da cella di memoria deve avere la possibilità di ricevere in ingresso un segnale binario e deve conservarne traccia in modo che alla sua uscita sia sempre possibile leggere quel valore, fino al momento in cui non viene modificato.
I circuiti di memoria sono detti *flip-flop*, termine intraducibile che indica la loro attitudine a commutare da uno stato a

Il piú semplice fra i possibili elementi di memoria, un bistabile o "flip-flop". Questo circuito è in grado di commutare da uno stato a un altro e di mantenersi nello stato assunto fino a che non riceve in ingresso un segnale opposto, conservando cosí "memoria" del segnale ricevuto.

un altro e a mantenersi nello stato raggiunto fino a nuovo ordine. Da un punto di vista funzionale, il piú semplice fra i tipi di flip-flop può essere visualizzato come una scatola nera con due ingressi, convenzionalmene indicati come S (per *set*) e R (per *reset*), e due uscite, indicate con Q e $\overline{Q}$ (che sono l'una il completamento dell'altra, cioè la "negazione" l'una dell'altra). Come si comporta questo flip-flop? Supponiamo che all'istante T1 Q sia a 1 (e automaticamente $\overline{Q}$ a 0). Basta un brevissimo impulso in arrivo sulla linea R perché l'uscita commuti: Q diventa 0, $\overline{Q}$ diventa 1. Un successivo impulso sulla linea R non modifica questa situazione; Q continua a fornire il valore 0. Basta invece anche un breve impulso sulla linea S perché la situazione muti nuovamente: Q torna a 1, $\overline{Q}$ va a 0, e in questa condizione qualunque impulso successivo sulla linea S non muta l'uscita. Un impulso sulla linea S porta a 1 (*set*) l'uscita; un impulso sulla linea R la riporta a 0 (*reset*).
Ambedue le lineee R e S, dunque, possono trasmettere uno 0 o un 1: il significato dei due impulsi binari è diverso nei due casi. Quello che abbiamo chiamato "impulso" è la trasmissione di un 1 logico (indipendentemente dai valori di tensione scelti effettivamente dal costruttore). Quando ambedue le linee R e S sono a 0, non succede nulla: l'uscita rimane quella che è. Se R è a 0 e S a 1, Q va a 1 (o rimane a 1 se già dava questo valore). Se R è a 1 e S è a 0, Q va a 0 (o rimane a 0 se già dava questo valore). Se tanto R quanto S sono a 1, l'uscita è indefinita: è una situazione da evitare, perché corrisponderebbe a chiedere alla cella di memoria di assumere contemporaneamente il valore 0 e il valore 1. Tutte queste cose possono essere riassunte, molto piú efficacemente di questa lunga descrizione a parole, in una particolare tavola di verità.
Questo tipo di flip-flop ammette una costruzione molto semplice, con l'uso di porte logiche OR e NOT, come si può vedere in figura: il "trucco" sta tutto nel rinviare l'uscita del circuito all'ingresso, in modo che la tensione in uscita si sostenga per tutto il tempo che il circuito è alimentato.
Il flip-flop R-S (come viene definito questo tipo di circuito di memoria) è molto semplice, ma presenta il meccanismo essenziale che sta alla base di tutti i tipi di circuiti bistabili: esistono infatti vari flip-flop piú complessi, ma il perfezionamento sta tutto sostanzialmente nel regolare meglio la sincronizzazione del passaggio della cella di memoria da un valore all'altro. Infatti nel flip-flop R-S che abbiamo visto, gli impulsi sulle linee R e S possono arrivare in qualunque istante, il che può rendere difficile tener conto del momento esatto in cui la cella di memoria assume un nuovo valore. In genere si usano invece flip-flop sincronizzati (*clocked flip-flop*) per i quali le linee R e S possono provocare una variazione nel contenuto della cella solo quando il *clock* (l'"orologio" sincronizzatore dell'unità di elaborazione nel suo complesso) invia un segnale di sincronizzazione. Fra un segnale di sincronizzazione e l'altro il contenuto della cella resta immutato. In questo modo è possibile sapere sempre esattamente quando avviene l'eventuale transizione da uno stato all'altro.
La sincronizzazione non è un problema solo per la memoria: se i tempi di esecuzione di tutte le attività di una serie complessa di circuiti fossero diversi e casuali, sarebbe impossibile avere la sicurezza del risultato. Un circuito potrebbe ricevere il segnale che attende in in-

gresso un po' prima che il circuito precedente abbia completato la sua funzione, oppure cosí in ritardo da perdere qualche informazione. Per evitare tutti questi inconvenienti, tutte le attività all'interno di una unità di elaborazione sono governate da un orologio sincronizzatore (anch'esso è un circuito elettronico), che invia impulsi a intervalli regolari. Ogni impulso segna un ciclo di attività: fra un impulso e l'altro non possono iniziare attività o fasi di attività. Cosí è sempre possibile prevedere con esattezza che cosa succede all'interno della macchina, a ogni singolo istante.

Quando i tasselli si incastrano

Circuiti di elaborazione, circuiti di memoria: abbiamo tutto l'armamentario di base del calcolatore, tutti i tasselli necessari per mettere insieme il mosaico. Seguire come si possano aggregare migliaia di questi elementi per realizzare i dispositivi che si possono trovare in commercio è solo questione di pazienza. A noi questo tuttavia non interessa: il nostro obiettivo è solo quello di capire meglio come funzioni al suo interno un personal computer, quindi possiamo passare a un livello superiore, quello delle unità funzionali che costituiscono l'unità di elaborazione. Possiamo ripartire dal semplice schema funzionale del calcolatore, che abbiamo introdotto all'inizio (vedi pagina 16): unità di ingresso – unità di elaborazione – unità di uscita. Quello che vogliamo fare è andare a vedere meglio come si articola l'unità di elaborazione: abbiamo visto quali funzioni svolga complessivamente, che cosa ci consenta di fare, quale sia il suo aspetto esteriore, e dall'altra parte quali siano i mattoni elementari con cui è costituita. Ci manca il livello intermedio.
All'interno dell'unità di elaborazione trovano posto quattro elementi fondamentali: il clock o orologio sincronizzatore, l'unità di controllo, l'unità di memoria e l'unità aritmetica e logica o ALU (*Arithmetic and Logic Unit*). Del clock abbiamo già parlato piú d'una volta: è un circuito molto semplice, che si limita a inviare impulsi a intervalli regolari, sui quali vengono sincronizzate tutte le attività che hanno luogo nell'unità di elaborazione. Anche l'unità di memoria è un elemento di semplice comprensione: è una matrice di celle in cui possono essere archiviate cifre binarie, le quali rappresentano dati o istruzioni. L'unità aritmetica e logica è quella in cui si svolgono le effettive operazioni di elaborazione: quella che contiene i circuiti per sommare o sottrarre numeri, effettuare operazioni logiche, incrementare o decrementare numeri, effettuare scorrimenti a sinistra o a destra, e simili. L'unità di controllo, infine, è deputata a controllare la successione delle operazioni di tutti gli altri componenti, in base alle istruzioni che riceve dal programma (archiviato in memoria). L'unità di controllo legge un'istruzione nella memoria, la decodifica e la trasforma in una serie di segnali di controllo per l'unità aritmetica e logica, in modo che questa segua le operazioni richieste sui dati opportuni. Di fatto, in genere l'unità di controllo, l'ALU e il clock sono ospitati in un unico chip: il microprocessore.
Come avvengono le comunicazioni fra questi diversi tipi di dispositivi e con l'esterno? Nella pratica costruttiva attuale, mediante un'unica struttura, che prende il nome di *bus* e che funziona un po' come un'autostrada, a cui tutte le singole unità hanno accesso e su cui possono far viaggiare i propri messaggi. Fisicamente il bus è costituito da una serie di linee di comunicazione (piste nei circuiti stampati, conduttori); sotto il profilo funzionale si articola invece in tre parti, che possono anche non essere fisicamente distinte. C'è innanzitutto il *bus dei dati*: su questo viaggiano le effettive informazioni che ciascun dispositivo deve comunicare ad altri. C'è poi il *bus degli indirizzi* su cui viaggiano solo le informazioni provenienti dall'unità di controllo e dall'ALU, che servono a selezionare il dispositivo (ed eventualmente la posizione all'interno del dispositivo) a cui debbono giungere le informazioni che verranno trasmesse sul bus dei dati. Infine sul *bus di controllo* transitano tutti i segnali necessari per definire il modo in cui deve avvenire il

Il modello fondamentale del calcolatore, visto un po' piú in dettaglio: il fondo in grigio scuro delimita i blocchi che sono ospitati nel microprocessore.

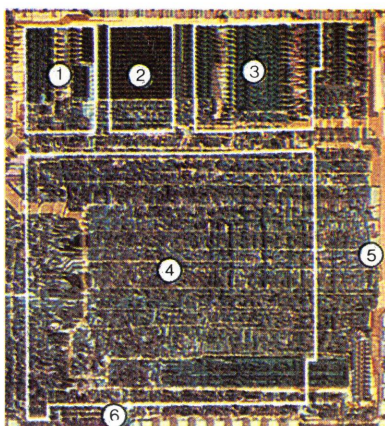
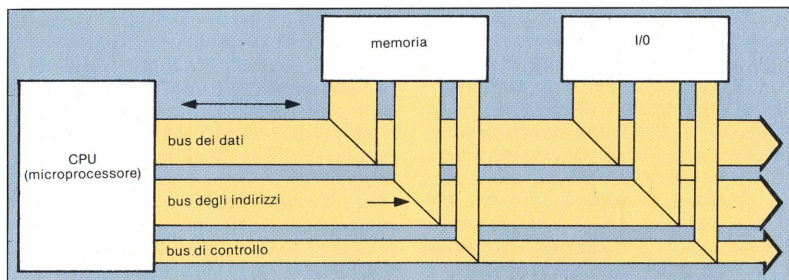
trasferimento delle informazioni fra le varie unità, modo che dipende dalle caratteristiche delle singole unità, dal tipo di informazioni che debbono essere comunicate, e dal verso in cui deve avvenire la trasmissione. I tre bus, abbiamo detto, non sono necessariamente distinti fisicamente: le stesse linee di trasmissione possono essere utilizzate in momenti diversi per inviare, per esempio, indirizzi e dati. In questo caso sul bus di controllo debbono essere inviati segnali che specifichino, istante per istante, quale è la funzione svolta, e sono necessari dispositivi per separare dati e indirizzi che viaggiano sulle stesse linee. Questi dispositivi si chiamano *multiplexer* o multiplatori e si dice che il bus dei dati è multiplexato con il bus degli indirizzi. Il numero di linee presenti in ciascun bus dipende dalle caratteristiche delle varie unità funzionali. Nel caso di un microprocessore a 8 bit, in cui cioè tutte le informazioni sono codificate in "parole" lunghe 8 bit (un byte), per esempio, il bus dei dati ha otto linee.

Il microprocessore

La figura di pag. 134 presenta lo schema di un microprocessore molto semplice: non è un microprocessore realmente in commercio, ma un dispositivo ideale, che possiede tuttavia le caratteristiche essenziali di un microprocessore. Accanto all'ALU e all'unità di controllo, il microprocessore incorpora una serie di dispositivi di memoria di piccole dimensioni: buffer o memorie di transito in cui vengono depositate temporaneamente informazioni in ingresso o in uscita, per adeguare le velocità operative delle unità interne al microprocessore alle velocità di trasferimento sui vari bus; e piccole memorie chiamate *registri*, che possono contenere in genere un byte, e che servono come deposito momentaneo di informazioni a cui l'ALU deve poter accedere molto velocemente. Fra questi registri vi è, in particolare, il registro *accumulatore* in cui vengono depositati i risultati delle operazioni compiute, prima del loro eventuale trasferimento a una posizione di archiviazione definitiva. Alcuni registri sono di uso generale: possono cioè essere sfruttati a piacere dai programmi, per facilitare lo svolgimento di operazioni par-

ticolari; altri registri, variabili da microprocessore a microprocessore, possono adempiere esclusivamente a funzioni particolari, per esempio per archiviare indirizzi di memoria. Quest'ultima particolarità può essere utile per realizzare forme di "indirizzamento" indiretto: il programma, per esempio, può comunicare all'ALU di eseguire una determinata operazione su dati conservati nella memoria, ma senza precisare l'esatta locazione di memoria in cui si trovano, dicendo invece che l'indirizzo della loro locazione è depositato in quei registri specializzati. Questo permette di usare indirizzi che sono stati determinati nel corso dello svolgimento del programma, e non potevano essere fissati a priori quando il programma è stato formulato. Un registro di uso speciale è quello indicato con la lettera F (che sta per *flag*, letteralmente "bandiera"). Questo registro può essere, per esempio, di un byte: ciascuna delle cifre binarie viene uti-

lizzata per conservare traccia di situazioni particolari. Una delle cifre, per esempio, può ospitare il riporto dall'ultima operazione eseguita, oppure può segnalare se nell'esecuzione dell'ultima operazione c'è stato uno straripamento (*overflow*): casi che abbiamo incontrato negli esempi di costruzione dei nostri circuiti elementari. Questo registro è usato ampiamente per definire condizioni di salto: un'istruzione del programma può fare riferimento a una posizione all'interno di questo registro per stabilire che l'esecuzione deve passare attraverso certe operazioni se si verifica una determinata condizione (per esempio se c'è stato straripamento nell'ultima operazione, e quindi se all'interno del registro F la cifra che corrisponde all'overflow è un 1). Questo registro ha quindi un ruolo molto importante, perché permette di dotare i programmi del calcolatore di quella flessibilità che deriva dalla possibilità di sequenze alternative di istruzioni.

L'ultimo registro è il contatore di programma o contatore delle istruzioni, nel quale viene conservato l'indirizzo dell'istruzione in fase di esecuzione. Quando l'esecuzione è terminata, questo registro è associato a dispositivi logici che attuano l'incremento di una unità dell'indirizzo: le istruzioni vengono archiviate in memoria in una serie di locazioni contigue. Quando l'esecuzione di un'istruzione è terminata, quindi, il contatore di programma viene incrementato di una unità: questo avvia tutto il processo gestito dall'unità di controllo, che va a recuperare l'istruzione archiviata nella locazione di memoria indicata dal

Un microprocessore, tipicamente, contiene gli elementi individuati nel disegno della pagina a fronte. I microprocessori non sono tutti uguali (differiscono profondamente l'uno dall'altro nei dettagli), ma gli elementi fondamentali sono sempre presenti. Qui sopra sono evidenziate alcune componenti di un microprocessore commerciale, lo Z8000 della Zilog: sono individuati i flag (1), l'unità aritmetica e logica o ALU (2), i registri (3), i circuiti logici (4), i controlli dei bus (5, 6). Un tipico sistema personale ha una struttura molto semplice, in cui il microprocessore (con le sue varie unità e in particolare con l'orologio sincronizzatore), la memoria e le unità di ingresso e uscita sono tutte collegate fra loro attraverso tre "piste", chiamate bus, su cui corrono i vari segnali: un bus dei dati, un bus degli indirizzi e un bus di controllo. Anche in questo caso si tratta di tre oggetti logicamente, ma non necessariamente anche fisicamente distinti: in alcuni sistemi lo stesso insieme di conduttori può portare a turno segnali di natura diversa.

contatore di programma, la invia all'ALU e ne determina l'esecuzione. I circuiti logici che determinano l'incremento del contatore di programma possono essere "saltati" dalle istruzioni di salto: istruzioni che dicono all'ALU di controllare se si verifica una certa condizione e, in caso di risposta positiva, di modificare l'indirizzo presente nel contatore di programma in modo che la successiva istruzione eseguita non sia quella seguente nel normale ordinamento numerico, ma un'altra istruzione, archiviata in una diversa posizione della memoria.

Il ciclo delle istruzioni

Quando vengono introdotte nel calcolatore, le istruzioni che costituiscono un programma vengono immagazzinate nella memoria interna (la RAM, memoria ad accesso casuale), a partire da una posizione prefissata. Da qui vengono eseguite in successione, a meno di istruzioni che specifichino altrimenti. Ciascuna istruzione, per essere eseguita, deve essere prima prelevata dalla RAM ed essere trasferita all'unità aritmetica e logica, dopo essere stata decodificata dall'unità di controllo. L'intervallo di tempo necessario perché un'istruzione venga trasferita dalla RAM al microprocessore, decodificata ed eseguita, viene definito *ciclo dell'istruzione*. Il ciclo dell'istruzione può avere una durata variabile, perché non è detto che tutte le istruzioni abbiano la stessa lunghezza (in effetti possono essere lunghe uno, due o tre byte). Il ciclo di un'istruzione può dunque occupare piú unità di tempo fissate dall'orologio sincronizzatore: si usa come unità di misura il *ciclo di macchina*, che ha una durata pari a tre impulsi dell'orologio. Un ciclo di macchina viene occupato dalla fase di prelevamento dell'istruzione dalla memoria: il contatore di programma invia l'indirizzo dell'istruzione sul bus degli indirizzi, e l'unità di controllo invia per parte propria un impulso che specifica come l'operazione richiesta sia un'operazione di lettura. L'istruzione richiesta viene di conseguenza inviata sul bus dei dati e arriva al registro delle istruzioni. Il decodificatore dell'unità di controllo decodifica questo primo byte, e stabilisce al contempo se si tratta dell'unico byte dell'istruzione o se l'istruzione comprende altri byte. In questo secondo caso, i cicli di macchina successivi sono spesi per il prelevamento dei byte ulteriori, che vengono depositati in altri registri (il registro delle istruzioni contiene solo il primo byte). Tutte queste operazioni vengono effettuate entro il secondo impulso d'orologio dell'ultimo ciclo di macchina richiesto dal numero di byte dell'istruzione. L'esecuzione dell'istruzione avviene poi nell'arco dell'ultimo impulso di quel ciclo di macchina se possibile; altrimenti è sempre compito dell'unità di controllo stabilire il numero di ulteriori cicli di macchina necessari per la fase di esecuzione. In particolare, saranno necessari altri cicli di macchina quando l'istruzione richiede il prelevamento dalla memoria di dati (che possono essere piú di uno): i dati debbono essere trasferiti in opportuni registri.
Benché i registri a loro volta non siano altro che celle di memoria, sono celle di memoria immediatamente accessibili per l'unità aritmetica e logica: fisicamente piú vicine all'ALU, per il microprocessore costituiscono piccole memorie privilegiate, caratterizzate da un tempo di accesso brevissimo.

Le istruzioni e la programmazione

Dal punto di vista del programmatore, molti dei dettagli che abbiamo visto non sono significativi. Per chi programma nel linguaggio specifico della macchina, l'unità di elaborazione appare semplicemente come un insieme di registri e una memoria principale. Di tutto il resto si prende cura automaticamente il microprocessore. Cosí un programma non deve dire qualcosa come "prendi l'indirizzo dell'istruzione successiva, mandalo al bus degli indirizzi, invia un segnale di lettura sul bus di controllo..." e via di questo passo, ma si limiterà, per esempio, a dire l'equivalente di "carica nel registro A i contenuti della cella di memoria XY". Anche la programmazio-

Istruzione	**Commenti**
ADD A, s; ADC A, s	Somma a 8 bit, o somma con riporto
SUB s; SBC A, s, CP s, NEG	Sottrazione a 8 bit, sottrazione con riporto, confronto e negazione dell'accumulatore
AND s	Operazioni logiche
OR s; XOR s	Incremento a 8 bit
INC s	Decremento a 8 bit
DEC m	Somma a 16 bit
ADD DD, ss	Somma a 16 bit con riporto
ADC HL, ss	Sottrazione a 16 bit con riporto
SBC HL, ss	Rotazione dell'accumulatore
RLA, RLCA, RRA, RRCA	Rotazione e shift della locazione m
RLD, RRD	Aggiustamento decimale dell'accumulatore
SCF	Set del flag di riporto
CCF	Complemento del flag di riporto
IN r, (C)	Input a registro in modo indiretto
CPI, CPIR, CPD, CPDR	Istruzioni di ricerca dei blocchi
LD A, I; LD A,R	Copia il contenuto dell'IFF (flip-flop di interrupt) nel flag P/V
BIT b, s	Copia il complemento del bit di locazione b nel flag Z
NEG	Negazione dell'accumulatore

La tabella presenta una parte (il "set") di istruzioni del microprocessore Z-80, uno dei piú diffusi nel mondo dei personal.

ne in linguaggio macchina, quindi, è semplificata da un insieme di automatismi incorporati all'origine nella macchina stessa.
Come sono fatte le istruzioni? Lo abbiamo già detto: sono seguenze di 0 e 1, esattamente come i dati e tutte le informazioni che circolano all'interno del calcolatore. E nulla le differenzia dai dati, se non la posizione che occupano nella successione dei byte che costituiscono il programma.
Rappresentare le istruzioni sotto forma di successioni di bit 0 e 1 è lungo, noioso, e la leggibilità del risultato è scarsissima. Per questo normalmente i programmatori usano rappresentare le istruzioni in notazione esadecimale: una cifra esadecimale equivale a quattro cifre binarie, e pertanto sono sufficienti due cifre esadecimali per codificare un byte. Cosí, per esempio, nel microprocessore Z-80 il numero esadecimale 3E, che corrisponde al binario 00111110, rappresenta l'istruzione "carica nel registro A". L'istruzione completa occupa due byte, in effetti: il secondo dà il numero che deve essere caricato nel registro. 3E 19 esadecimale rappresenta allora l'istruzione "carica nel registro A il numero 25" (19 esadecimale corrispondente a 25 decimale: $1 \times 16 + 9 \times 1$). Va notato che allora 3E 3E, cioè la successione dei due byte binari 00111110 00111110, significherà "carica nel registro A il numero 62 decimale": ecco un caso di due byte del tutto identici che vengono interpretati in modo nettamente diverso, il primo come codice di una operazione (*codice operativo*) e il secondo come codice di un numero (*operando*). In un'istruzione che occupa piú byte il primo è sempre interpretato come codice operativo, i successivi come operandi.
Un'altra istruzione del microprocessore Z-80 ha il formato D6 n (dove n sta al posto di un numero qualsiasi), e ha il significato "sottrai il numero n dai contenuti del registro A". Potremmo dunque immaginare questa minuscola successione di istruzioni:

3E	3E
D6	76

che significherebbe “carica nel registro A il numero 62; poi sottrai 25 dal contenuto di A”. Si tratta, ovviamente, solo di un esempio banale. Se per caso, però, ci dimenticassimo di scrivere uno dei due 3E, che cosa succederebbe? La macchina vedrebbe in realtà la successione 3E D6 19, prenderebbe 3E correttamente per un codice operativo ma poi, avendo riconosciuto in 3E il codice operativo di un'istruzione da due byte, va a interpretare il D6 che segue non piú come codice operativo di un'altra istruzione, ma come codice di un numero: e caricherebbe nel registro A il numero 214 (D è 13 decimale, quindi D6 = 13 × 16 + 6 × 1 = 208 + 6 = 214). Il 76 esadecimale, che volevamo fosse interpretato come codice di un numero, a questo punto verrà interpretato invece come codice operativo di una nuova istruzione. Nello Z-80 il 76 esadecimale come codice operativo corrisponde all'istruzione HALT, che blocca l'unità di elaborazione. In questo caso, dunque, il nostro programma si bloccherebbe; in altri casi, per un errore analogo, potremmo trovarci risultati del tutto inattesi.
E qui ci fermiamo, per quanto riguarda la programmazione in linguaggio macchina: riprenderemo il tema della programmazione e dei linguaggi di programmazione nel prossimo capitolo.

Cablato e programmato

Le operazioni elementari che è in grado di svolgere un microprocessore sono effettivamente elementari: come abbiamo visto, spostare bit da una parte all'altra, sommare, controllare, confrontare e simili. Sta tutto all'abilità del programmatore riuscire, sulla base di queste operazioni semplicissime, costruire programmi in grado di svolgere funzioni complesse come, per esempio, memorizzare una serie di parole e metterle in ordine alfabetico oppure tutte le funzioni connesse all'elaborazione di un testo o alla gestione di una base di dati. Nulla, tuttavia, impedirebbe teoricamente di costruire circuiti di complessità molto superiore, che svolgessero funzioni altrettanto piú complesse. Nei microprocessori, per fare solo un esempio, non esiste una operazione elementare di moltiplicazione: nei calcolatori piú potenti, invece, spesso esiste. Il vantaggio è notevole, in termini di velocità.
Le funzioni che sono fornite direttamente dall'hardware della macchina si dicono funzioni *cablate*; quelle che si ottengono per combinazione delle funzioni elementari, tramite un programma fornito dall'esterno, sono funzioni *programmate*. Il confine tra i due tipi di funzioni varia, dunque, da macchina a macchina, e può essere un fattore importante per talune applicazioni speciali.
Per compiti specifici, che richiedono alta velocità di esecuzione, si può pensare di “specializzare” i circuiti di una macchina. Un microprocessore di uso generale, per esempio, potrebbe essere impiegato benissimo per controllare un elettrodomestico come una lavabiancheria o un'apparecchiatura ad alta fedeltà: basterebbe collegarlo opportunamente ai dispositivi che deve controllare e fornirlo, volta a volta, del programma adatto. Sarebbe però un modo di procedere molto scomodo: in effetti si preferisce utilizzare, in tutti questi casi, un microprocessore che svolga specificamente quelle funzioni. La logica di funzionamento – il programma – in questi casi è cablata: i circuiti sono predisposti in modo da svolgere esattamente quelle funzioni, e per questo sono piú veloci, piú precisi e affidabili. È normale, però, che nella fase di progettazione si lavori con microprocessori di uso generale, fino a che non è stato messo a punto il programma adatto: si passa alla logica cablata in una seconda fase.
Un caso illuminante è quello delle macchine che giocano a scacchi. Gli scacchi sono un gioco complesso, e per ogni posizione di gioco si prospettano un gran numero di sviluppi successivi possibili: se la macchina deve esaminare

tutti gli sviluppi interessanti per la profondità anche solo di poche mosse successive, si arriva presto a migliaia di sequenze di mosse, che debbono poi essere valutate e confrontate, perché la macchina possa decidere come comportarsi. A parte i problemi legati alla memorizzazione di quantità cosí elevate di dati, c'è un problema di velocità: la macchina deve decidere nel giro di qualche minuto, non di qualche giorno. Un programma che giri su una macchina di uso generale può essere molto lento nello svolgimento di questo compito: per ottenere risultati migliori, alcuni fra i ricercatori attivi in questo campo hanno costruito macchine "dedicate", in cui cioè la logica del compito da svolgere viene incorporata direttamente nell'hardware, grazie a circuiti progettati espressamente a questo scopo. In questo modo la macchina è piú veloce e – a parità di tempo – può giocare meglio di una macchina in cui la logica del gioco sia programmata.
Il pregio della logica cablata è la velocità, mentre il pregio della logica programmata sta nella versatilità. Se si deve cambiare qualcosa, nel caso di un programma non ci sono grossi problemi; nel caso di un dispositivo fisico deputato a svolgere una funzione ben precisa, se cambiano le necessità non si può modificare nulla, ma bisogna sostituire integralmente il dispositivo.
Per ovviare a questo inconveniente, ma avere tutti i vantaggi di una logica cablata, si può impiegare una forma di *microprogrammazione*: si tratta tuttavia di una tecnica ancora poco sviluppata, impiegata solo su alcune macchine particolari e di grandi dimensioni, ma è probabile che il suo campo di applicazione vada allargandosi nel futuro prossimo, fino a toccare anche i calcolatori personali. Come un programma combina le operazioni elementari di cui è capace la macchina, per ottenere funzioni complesse, la microprogrammazione consentirebbe di organizzare i dispositivi fisici elementari in modo da ottenere diversi tipi di funzioni elementari. Anche la struttura interna dell'unità di elaborazione, cioè, potrebbe essere programmata: i circuiti interni non sarebbero fissi, ma potrebbero essere modificati e rimodificati per ottenere, volta a volta, insiemi di istruzioni elementari diversi. Certo la microprogrammazione non è impresa da poco, e richiede una buona conoscenza dell'hardware, ma non è escluso che in futuro si trovino metodi semplici per mettere anche questa possibilità alla portata di tutti.

Una EPROM è una memoria a sola lettura cancellabile (irraggiandola con luce ultravioletta attraverso una finestrella) e programmabile. Sotto, un programmatore di EPROM: i dati possono essere inseriti da tastiera o trasferiti dal calcolatore.

RAM, ROM, PROM, EPROM

Un modo piú semplice per incorporare un programma nell'hardware del calcolatore sta non nel modificare i circuiti di elaborazione stessi, ma nell'inscrivere in modo indelebile nella memoria il programma. La memoria di cui abbiamo parlato nelle pagine precedenti era sempre una memoria RAM, cioè una memoria in cui è possibile sia leggere sia scrivere. Per rendere piú efficiente l'uso di un programma, lo si può invece registrare in una ROM, una memoria a sola lettura (in cui non si possono piú

scrivere informazioni, dopo la scrittura iniziale). Il vantaggio di una ROM è che, non dovendo ospitare informazioni variabili, ma sempre le stesse informazioni, i suoi circuiti possono essere piú semplici; la RAM disponibile, inoltre, viene lasciata completamente libera per i dati su cui il programma deve lavorare e per i risultati. Le ROM sono fornite direttamente dal costruttore della macchina, di norma, e incorporano quei programmi di uso generale che si prevede siano indispensabili (o quasi): come abbiamo già visto nel capitolo precedente, i programmi di gestione fondamentale delle risorse della macchina, spesso anche il programma traduttore del BASIC. Nelle macchine portatili, per semplificare la vita all'utente, si possono trovare su ROM anche programmi applicativi diversi, tipicamente quelli per l'elaborazione di testi, gli archivi di dati, i tabelloni elettronici, le comunicazioni. Rispetto alla RAM, la ROM non perde piú i suoi contenuti, neanche quando viene tolta l'alimentazione.
Le ROM che si trovano nel personal computer al momento dell'acquisto sono memorie di sola lettura in cui i dati sono stati iscritti dal fabbricante al momento della produzione del dispositivo stesso; esistono però anche ROM differenti, in cui le informazioni possono essere scritte *dopo* la fabbricazione del dispositivo. Sono le PROM, o ROM programmabili: memorie a sola lettura in cui l'utente può scrivere – ma una sola volta – i suoi programmi. Il dispositivo uscito dalla fabbrica è in grado di accettare un programma: ma una volta scritto il programma, questo vi rimane impresso indelebilmente, e l'operazione di scrittura non può essere ripetuta. (La programmazione di una PROM richiede apparecchiature speciali.)
Esiste anche un terzo tipo di ROM: una memoria a sola lettura che, oltre a essere programmabile, è anche cancellabile (e quindi rende possibile piú volte la scrittura). La cancellazione si effettua esponendo la EPROM a luce ultravioletta, grazie alla presenza di una "finestra" nell'involucro che contiene e protegge il chip. Anche le EPROM debbono essere programmate mediante delle speciali apparecchiature (i *programmatori di EPROM*).

Le interfacce

I vari componenti che costituiscono un personal computer, e la sua unità centrale di elaborazione in particolare, sono separati fra loro e lavorano in modi diversi l'uno dall'altro: per questo è necessario che esistano circuiti appositi che provvedano a omogeneizzare i segnali che intercorrono fra i vari componenti e organizzare nel tempo gli eventi di comunicazione. Il problema interessa tutti i componenti, ma dei dispositivi interni all'unità centrale si occupa sempre il costruttore: le interfacce fra il microprocessore e la memoria RAM o ROM, per esempio, sono già incorporate nella macchina e l'utente non ha mai occasione di apprezzarne l'attività. Piú interessanti sono invece le interfacce fra l'unità centrale e i dispositivi esterni, dispositivi che possono essere di qualunque natura e spesso sono aggiunti alla configurazione del calcolatore in un secondo tempo. Il problema dell'interfacciamento tocca allora da vicino l'utente, il quale deve preoccuparsi della compatibilità fra i vari elementi che vanno a integrare la macchina.
I dispositivi esterni hanno modalità di funzionamento, in genere, diverse da quelle del microprocessore interno alla macchina: un caso tipico è quello delle stampanti. La velocità a cui una stampante stampa i caratteri è diversa dalla velocità a cui il microprocessore è in grado di inviare i dati: la stampante è molto lenta, per gli standard del microprocessore. L'interfaccia quindi deve poter ricevere i dati dal microprocessore alla velocità piú opportuna per quest'ultimo, ma deve poi inviarli alla stampante alla velocità opportuna per questa. L'interfaccia, poi, deve anche far sí che i segnali in uscita siano quelli adatti per il dispositivo che si deve pilotare: può quindi svolgere anche una funzione di transcodifica dei segnali che riceve, decodificandoli e ricodificandoli nella forma richiesta dal dispositivo esterno.

A tutto questo si deve aggiungere che l'interfaccia deve poter in qualche modo controllare la disponibilità dell'apparecchiatura esterna per l'operazione che si vuol farle compiere, e deve poter segnalare all'apparecchiatura esterna quando sta per iniziare l'invio di segnali da parte del microprocessore; oppure, in alternativa (se non si tratta di un dispositivo di uscita ma di un dispositivo di ingresso), deve poter richiedere al microprocessore la disponibilità ad accettare il flusso di informazioni in arrivo e deve segnalare l'inizio dell'invio delle informazioni stesse.
Possono esistere dunque tanti tipi di interfacce quanti sono i possibili dispositivi esterni, ma esistono alcuni principi di funzionamento comuni e, inoltre, sono state sviluppate alcune interfacce standard, alle quali molto spesso i costruttori si adeguano, rendendo relativamente facile l'interconnessione fra apparecchiature diverse.
Si può tracciare una distinzione generale, riguardo al modo in cui i segnali vengono trasferiti, tra interfacce *seriali* e interfacce *parallele*. Poiché dall'una parte come dall'altra dell'interfaccia i dispositivi operano normalmente non su bit singoli ma su gruppi di bit, cioè su byte o su parole di piú byte alla volta, un modo di trasferimento che si propone spontaneamente è quello di usare tante linee quanti sono i bit della parola con cui lavorano i dispositivi, e trasferire contemporaneamente un bit su ciascuna linea: è il funzionamento in parallelo. Molte interfacce, però, usano una sola linea di trasmissione, sulla quale inviano un solo bit alla volta: è il modo di trasferimento seriale.
Nel caso delle interfacce di tipo seriale, è necessario a monte e a valle qualche dispositivo di conversione che permetta il passaggio dalla (e alla) configurazione parallela utilizzata all'interno sia della sorgente, sia del destinatario. Il meccanismo piú comune è un registro a scorrimento: il byte che si deve trasmettere, per esempio, viene inviato al registro di scorrimento, al quale poi si applica una serie di operazioni di scorrimento in una direzione, verso sinistra per esempio.
Come abbiamo visto precedentemente, a ogni operazione di scorrimento viene inserito da destra un bit 0, mentre il bit piú a sinistra straripa ed esce. Qui, anziché andare a una cella di memoria che segnali l'avvenuto *overflow*, il bit può essere mandato alla linea di trasmissione. Con otto operazioni di scorrimento, un byte può essere inviato all'esterno in modo seriale.
Per i dati in arrivo, si può procedere in modo analogo, con un altro registro a scorrimento, in cui i bit possono essere inseriti da destra, facendo scorrere ogni volta i dati contenuti in precedenza verso destra. Dopo otto operazioni di questo tipo, il registro ospiterà un byte esatto di dati, trasmesso serialmente, ma ricostruito nella sua forma parallela.
Esistono interfacce sia seriali, sia parallele, per le stampanti: fra i tipi paralleli uno si è affermato in special modo nel campo dei personal computer, ed è l'interfaccia Centronics, che prende il nome dalla prima azienda che l'ha utilizzata e ha poi saputo imporla come standard di fatto. L'interfaccia seriale usata, invece, è un vero standard ufficiale, l'interfaccia che prende il nome di RS-232, e che è un'interfaccia generale di comunicazione. L'interfaccia RS-232 viene utilizzata, infatti, non solo con le stampanti, ma in tutte le forme di comunicazione a distanza, da computer a computer collegati direttamente da un cavo, oppure mediante linea telefonica (e dispositivi MODEM, di modulazione e demodulazione, in grado di convertire il segnale fornito dall'interfaccia in forma opportuna per la trasmissione su linea telefonica e, all'altro capo, di riconvertirlo nella forma richiesta dall'interfaccia stessa).
Tipi particolari di interfacce sono i convertitori da analogico a digitale (A/D) e da digitale ad analogico (D/A), a cui abbiamo avuto già occasione di far cenno: sono interfacce che trasformano segnali analogici, cioè continui, nei segnali digitali, discreti, richiesti dal calcolatore o, viceversa, trasformano i segnali digitali del calcolatore in segnali analogici di tipo opportuno, adatti per pilotare un dispositivo esterno.

1234567890
1234567890
1234567890
1234567890
1234567890

Dal codice binario al linguaggio naturale

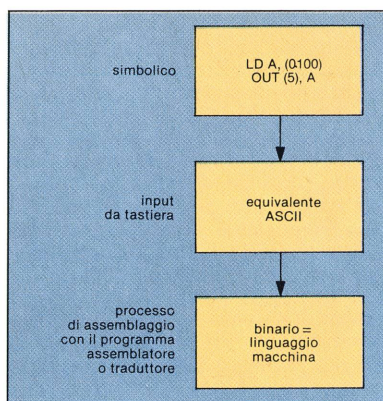
Un computer (e un personal come caso particolare), come abbiamo visto, è una macchina complessa, tuttavia la sua complessità deriva tutta dalla combinazione di unità elementari molto semplici, porte logiche, celle di memoria e via dicendo, organizzate in unità via via piú complesse. Quelle unità fondamentali *capiscono* in realtà un linguaggio elementare, costituito da due sole entità simboliche (il passaggio di corrente o meno; la presenza di una carica elettrica o meno) e dalle loro possibili combinazioni. Il passaggio o l'assenza di corrente, per comodità, possono essere rappresentati con i simboli 1 e 0: tutto il problema di costruire un'interfaccia ottimale fra la macchina e il suo utente umano sta nella capacità di costruire, a partire da questi due soli elementi del repertorio di base, strumenti di comunicazione agili ed efficaci.
Il sistema decimale che usiamo per far di conto è solo uno dei tanti modi possibili per rappresentare i numeri: senza andare molto lontano, usiamo invece un sistema misto per secondi, minuti, ore e giorni. Abbiamo visto come sia possibile costruire una notazione per i numeri utilizzando solo due cifre: il sistema binario. Ma sappiamo anche che è possibile codificare le lettere dell'alfabeto e i segni di interpunzione con i numeri: tanto per fare un esempio, si può far corrispondere 1 alla A, 2 alla B, 3 alla C e via di seguito. Quel che è esprimibile con numeri decimali è esprimibile in binario: anche per le lettere dell'alfabeto e i segni di interpunzione si può dunque formulare un codice sulla base di 0 e 1 solamente. I caratteri si possono combinare in parole: cosí sequenze opportune di 0 e 1 possono rappresentare qualunque comando o istruzione.
È questa la base del cosiddetto *linguaggio macchina*: è il linguaggio per parlare direttamente ai dispositivi fisici interni del calcolatore. Tutto quello che serve è una semplice interfaccia, che idealmente possiamo immaginare costituita da due soli pulsanti: sul primo è riportata la cifra 0, sull'altro la cifra 1. Il dispositivo non deve far altro che trasformare le cifre che noi digitiamo in opportuni impulsi di corrente.
Quali sono le sequenze di 0 e 1 da usare? Dipende strettamente dall'hardware che abbiamo a disposizione, e in particolare, parlando specificamente di personal computer, dal microprocessore utilizzato. Il microprocessore è costituito da una miriade di circuiti integrati predisposti in modo tale da effettuare alcune operazioni fondamentali quando ricevono in ingresso determinate sequenze di 0 e 1. Queste sequenze costituiscono il *set di istruzioni* (ovvero l'insieme di istruzioni) del microprocessore. Per programmare in linguaggio macchina, bisogna conoscere il set di istruzioni del microprocessore con cui si lavora. Se cambia il microprocessore, cambiano anche le istruzioni. Molte operazioni fondamentali saranno simili (tutti i microprocessori hanno un'istruzione di somma per numeri binari, un'i-

Riusciremo un giorno a interagire e comunicare con il calcolatore attraverso il linguaggio che usiamo tutti i giorni?

struzione di scorrimento a destra di una cifra e simili), ma magari codificate con sequenze binarie differenti.
Ma poi bisogna inserire come sequenze di cifre binarie anche i numeri che si vogliono elaborare, i caratteri e via dicendo... e come fa il microprocessore a distinguere le sequenze che rappresentano istruzioni da quelle che rappresentano numeri, da quelle che rappresentano caratteri e cosí via? Il microprocessore e tutti gli altri dispositivi fisici che costituiscono la struttura del calcolatore, in effetti, non distinguono proprio nulla: "vedono" solamente impulsi di corrente e in risposta a questi si pongono in uno stato piuttosto che nell'altro. L'interpretazione sta tutta a noi: il costruttore ha predisposto la macchina in modo tale che assuma un certo stato in risposta a una certa sequenza di impulsi binari: a quel punto la macchina risponderà in uno fra un certo numero di modi ben specificati alla successiva sequenza di impulsi binari. Semplificando un po', con un esempio: una certa sequenza binaria ha portato il microprocessore in uno stato *S* che, per il modo in cui il costruttore ha organizzato le varie unità fondamentali, è uno stato di attesa di due sequenze di impulsi binari, che poi verranno sommati secondo le regole della somma binaria. Le due sequenze successive da immettere, dunque, vanno interpretate come codici di numeri.
Le operazioni elementari della macchina a cui si accede attraverso il set di istruzioni del microprocessore sono veramente elementari: in genere non esiste nemmeno la moltiplicazione di numeri binari! Quindi chi programma in linguaggio macchina deve fare un lavoro davvero da certosino: deve tener conto con estrema precisione di tutto quello che succede internamente, deve dire al microprocessore che cosa deve fare in un dettaglio totale. Non può limitarsi a dire "memorizza il numero tal dei tali"; deve proprio specificare "memorizza il numero tal dei tali nella cella di memoria tal dei tali". E quando chiede alla macchina, in seguito, di memorizzare un'altra cosa, deve ricordarsi dove ha messo la prima, per evitare di cancellarla sovrapponendovi una nuova sequenza binaria.
Il linguaggio macchina ha uno svantaggio evidente: è troppo lontano dal nostro modo di parlare, di formulare i problemi e gli schemi di soluzione. Inoltre è noiosissimo, e commettere errori nella scrittura di un programma in linguaggio macchina è facilissimo, mentre è poi difficile riuscire a identificarli. A controbilanciare gli svantaggi sta il vantaggio di avere un controllo completo delle caratteristiche fisiche della macchina, e quindi di poterla far lavorare con la massima efficienza possibile.
Tuttavia la flessibilità del calcolatore elettronico programmabile è davvero splendida. Perché, invece di continuare a scrivere programmi in linguaggio macchina, non scrivere un programma "traduttore", che in qualche modo accetti in ingresso un linguaggio piú adatto al nostro modo di comunicare e poi lo traduca automaticamente in linguaggio macchina? L'idea è venuta subito ai primi costruttori di calcolatori ed è stata un'idea semplice ma brillante. È possibile, in effetti, scrivere programmi traduttori che accettano in ingresso espressioni piú vicine al nostro modo di parlare e le traducono in opportune sequenze di cifre binarie. Il linguaggio con cui potremo allora colloquiare con il calcolatore sarà piú comodo: dovrà comunque essere rigido, perché il programma deve poter prevedere tutte le possibilità (altrimenti come farebbe a tradurre?), tuttavia il nuovo linguaggio potrà essere tanto ricco quanto sapremo costruirlo.
Il compito di scrivere un programma traduttore di questo genere non è affatto facile: è un compito per pochi specialisti. E anche le tecniche per scrivere programmi traduttori sono ancora in evoluzione: i primi programmi traduttori erano relativamente semplici, rispetto a quello di cui disponiamo oggi, e nascono continuamente programmi traduttori con caratteristiche differenti, sempre piú evoluti o specializzati.
Il programma traduttore, naturalmente, non è il linguaggio: il linguaggio è il sistema di simboli che usiamo noi utenti

per colloquiare con la macchina, attraverso la tastiera. Il programma traduttore deve ricevere in ingresso questi simboli, deve poterli riconoscere (e deve poter riconoscere sequenze di simboli che non costituiscono parole o frasi grammaticalmente corrette) e quindi deve poter associare loro delle opportune sequenze di simboli nel linguaggio interno della macchina.
Se volete divertirvi a lambiccarvi un po' il cervello, dovete tener presente che comunque il programma traduttore è scritto in linguaggio macchina (per partire dalla base) e che comunque deve ricevere in ingresso a sua volta sequenze di cifre binarie. Quello che arriva al programma traduttore non è mai, neanche in questo caso, la lettera "A" o la lettera "B": appena premiamo un tasto sulla tastiera del calcolatore, una serie di commutatori provoca l'invio di una sequenza di impulsi di corrente all'unità centrale. Subito le informazioni vengono codificate in una successione di 0 e 1! Il codice tuttavia è diverso: questo è il codice per comunicare con il programma traduttore, che lo riceve, lo analizza, lo interpreta e lo trasforma in un nuovo codice – quello adatto per la macchina. Il principio, in linea di massima, è abbastanza semplice: ma riuscire a seguirlo in dettaglio è tutt'altro genere di impresa.

Alto e basso livello

Il linguaggio macchina è il linguaggio *di basso livello* per eccellenza: è il piú legato alle caratteristiche fisiche dell'hardware. Il *livello* di un linguaggio si alza a mano a mano che il linguaggio stesso si distanzia dal codice macchina e dalle caratteristiche fisiche dell'hardware. Un linguaggio *di alto livello* cerca di alleviare il compito del programmatore offrendogli, per comunicare con la macchina, uno strumento che abbia caratteristiche di maggiore maneggiabilità per l'uomo. Il linguaggio di livello *piú alto possibile* sarebbe quello che usiamo tutti i giorni: un traguardo comunque ancora molto lontano. Per il momento i linguaggi di programmazione offrono vari livelli di avvicinamento a questo ideale, usando parole del linguaggio ordinario (dell'inglese, quasi sempre) e una sintassi che ha qualche vaga rassomiglianza con quella di una lingua scritta, ma è molto piú rigida e precisa. Si può dire che il livello di un linguaggio di programmazione è tanto piú alto, quanto piú ci consente di colloquiare con la macchina usando un linguaggio vicino a quello umano.
Il linguaggio di alto livello ci libera, almeno in una certa misura, dalla difficoltà di maneggiare sequenze di 0 e 1, e ci permette di essere indipendenti dalle caratteristiche fisiche della macchina. È il programma traduttore che si incarica, per conto nostro, di gestire le risorse della macchina. Non dobbiamo piú dire "memorizza il numero tal dei tali nella cella di memoria tal dei tali": a questo punto è sufficiente dire "prendiamo il numero tal dei tali". Il programma traduttore ci tiene, a questo punto, la "contabilità" della memoria centrale: tiene conto delle celle di memoria libere e di quelle occupate, stabilisce dove sia meglio andare a memorizzare volta per volta quello che vogliamo.
Se il linguaggio non deve tener conto di questi particolari, strettamente connessi alla struttura fisica della macchina, questo significa che, passando da un microprocessore all'altro, da un calcolatore all'altro, le mie frasi espresse nel linguaggio "di alto livello" possono rimanere identiche: è sufficiente avere un nuovo programma traduttore, che traduca nel linguaggio macchina del nuovo hardware.
È questa la caratteristica della *trasferibilità*: un programma scritto in un linguaggio di alto livello può essere trasferito da una macchina all'altra, cioè può girare su qualsiasi macchina, purché per essa sia stato approntato un programma traduttore. Il programma traduttore, in linea di massima, non è invece trasferibile: esso è invece fondamentalmente legato alla struttura fisica della macchina. Tuttavia è pensabile operare traduzioni in piú stadi: costruire cioè un programma traduttore per un programma traduttore. Dal linguaggio ad alto livello si passa dunque, con un

Un programma in linguaggio di assemblatore ha una struttura vicina al linguaggio macchina: le sue istruzioni sono in corrispondenza biunivoca con quelle fondamentali della macchina. Qui si vede la sequenza di traduzione di un programma in assemblatore, mediata dal codice ASCII.

primo programma traduttore, a un codice intermedio, piú semplice (dal punto di vista dell'hardware), che poi viene tradotto da un secondo programma traduttore in linguaggio macchina. Il linguaggio traduttore di livello intermedio, in questo modo, può essere relativamente indipendente, a sua volta, dall'hardware. Questa soluzione, tuttavia, non è molto comune.

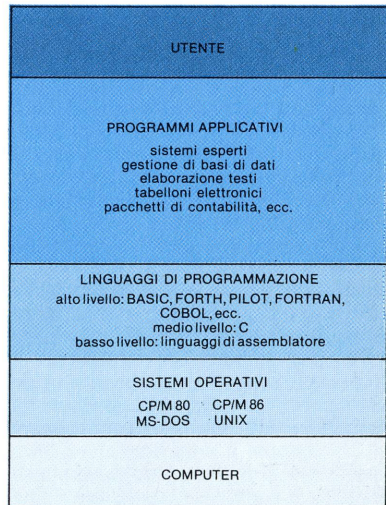
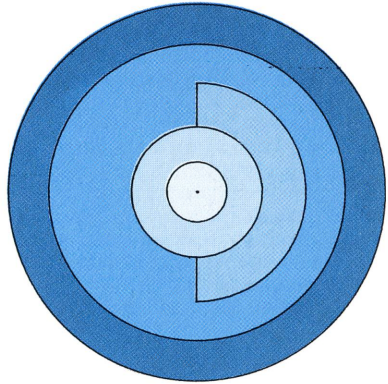
L'assemblatore

Il primo passo è quello di sostituire alle sequenze di 0 e 1 parole o sigle che ce ne ricordino il significato. Queste sigle sono dette *mnemoniche* e sono del tipo LDA (da *load accumulator*, cioè "carica nell'accumulatore") o ADD ("somma"). La corrispondenza fra il linguaggio macchina e questo codice di livello superiore è comunque biunivoca: il programmatore è facilitato perché può lavorare con simboli piú vicini a quelli della sua lingua, ma resta comunque vincolato alla struttura della macchina (deve continuare a preoccuparsi dei registri fisici dell'elaboratore, delle esatte locazioni di memoria).

Questo tipo di linguaggio è il *linguaggio di assemblatore* o *linguaggio assembly*. *Assemblatore* è invece il tipo di programma traduttore che si deve realizzare per effettuare la traduzione. La traduzione, tuttavia, è relativamente semplice, grazie alla corrispondenza biunivoca che esiste fra istruzioni nel linguaggio macchina e istruzioni nel linguaggio di assemblatore.
Il pregio dei linguaggi di assemblatore è quello di lasciare al programma il controllo completo delle risorse fisiche della macchina, alleggerendo tuttavia il carico della scrittura dei programmi permettendo di usare simboli un po' piú facili da maneggiare. Il ricorso al linguaggio di assemblatore è molto comune, nel caso dei programmatori di professione, per le situazioni in cui la velocità di esecuzione è un fattore determinante.

Interpreti e compilatori

Il linguaggio di assemblatore è ancora un linguaggio di "medio livello": piú facile, non ancora indipendente dalla macchina. I linguaggi di alto livello ci liberano dalla dipendenza dalla macchina e ci fanno respirare un po' di aria di casa: permettono di dare nomi significativi agli oggetti che si maneggiano, le istruzioni fondamentali sono piú vicine al nostro modo di affrontare i problemi che al modo di lavorare della macchina. Non si parla piú di registri e celle di memoria, ma di matrici o di liste di dati, di moltiplicazioni e di elevamento a potenza.
Ora non c'è piú corrispondenza biunivoca tra istruzioni nel linguaggio di alto livello e istruzioni nel linguaggio macchina: a una istruzione di alto livello possono corrispondere gruppi anche notevoli di istruzioni di macchina. Il compito del programma traduttore è decisamente piú impegnativo.
Per i linguaggi di alto livello esistono due categorie ben distinte di programmi traduttori: gli interpreti e i compilatori. I quali lavorano in modo molto differente gli uni dagli altri. Per conservare l'analogia della traduzione, gli interpreti si comportano proprio come gli interpreti simultanei, che traducono quel che vie-

ne detto da una persona frase per frase; mentre i compilatori si comportano come il traduttore di libri, che riceve un libro completo per la traduzione e può organizzarsi il lavoro come meglio crede, libero di leggere tutto prima di cominciare a tradurre, o di cominciare a tradurre dall'ultimo capitolo e via dicendo.
Un programma interprete lavora frase per frase: si mette all'opera, piú precisamente, non appena l'utente gli segnala la fine di un'istruzione (semplice o complessa) con un RETURN. Però, lavorando cosí "alla giornata", non ha modo di accorgersi se una certa istruzione si presenta cento volte: la ritradurrà da zero ogni volta che la incontra.
Un programma compilatore, invece, lavora sempre su programmi complessi e può tener conto anche della loro struttura complessiva: può rendersi conto che un'istruzione viene ripetuta cento volte, può tradurla la prima volta e poi "ricopiare", per cosí dire, quella traduzione senza ripensarla ogni volta. Non solo: lavorando sul programma completo, può anche essere dotato di qualche meccanismo per cercare di rendere il piú efficace possibile il codice macchina risultante, per renderlo compatto e veloce.
Un interprete permette di lavorare in modo immediato, non solo in modo differito, e rende possibile l'interattività fra il programmatore e la macchina: l'interprete segnala immediatamente un errore e la correzione può essere contestuale. Il programma, almeno entro certi limiti, può anche essere costruito interattivamente a pezzetti, controllati volta per volta.
Nel caso di un compilatore, invece, l'interattività non c'è. Si deve prima scrivere il programma per intero, con l'ausilio di un *text editor* o di un vero e proprio programma di word processing, come quelli di cui abbiamo parlato nel capitolo 1. Questa versione del programma prende il nome di *programma sorgente*. Quando il programma sorgente è completo e soddisfacente, si chiama in causa il compilatore e lo si fa compilare: il risultato della compilazione è un *programma oggetto* in codice macchina.

Fra l'hardware di un computer e l'utente finale si interpone una serie di livelli di software, ciascuno dei quali costruito con gli strumenti messi a disposizione dai livelli inferiori, e sempre piú lontano dal funzionamento effettivo dei componenti fisici. Il primo livello è costituito dai sistemi operativi, che gestiscono tutte le risorse della macchina; i programmi "traduttori" (interpreti e compilatori) per linguaggi di programmazione sfruttano i mezzi del sistema operativo, e i programmi applicativi, scritti in linguaggi di alto livello, "istruiscono" la macchina a svolgere compiti specifici, utili all'utente.

I compilatori in genere non svolgono tutta la loro attività in un solo passaggio: i passaggi intermedi danno luogo a versioni intermedie del programma. Dal punto di vista funzionale, il compilatore è composto da almeno tre sezioni: un *analizzatore lessicale*, a cui spetta il compito di identificare le singole "parole" contenute nel programma; un *analizzatore sintattico*, a cui spetta il compito di identificare la struttura del programma, cioè le relazioni sintattiche fra le varie parole identificate dall'analizzatore lessicale (l'analizzatore sintattico, cioè, deve svolgere l'equivalente di una "analisi logica" delle frasi del programma); l'ultima sezione è il *generatore di codice* vero e proprio che effettua la codifica in linguaggio macchina.
L'analizzatore sintattico o l'analizzatore lessicale incorporano le regole del linguaggio che devono compilare: possono identificare una parola o la struttura di una frase solo se rispettano alla lettera quelle regole. Dove trovano qualcosa che non può essere spiegato in alcun modo sulla base delle regole, si bloccano e segnalano un errore.
C'è un errore? La compilazione si interrompe, e arriva un segnale al programmatore: qualcosa non va. Non tutti i linguaggi (e non tutti i compilatori) si comportano nello stesso modo: alcuni "sanno" essere piú precisi di altri. Il compilatore segnala di essersi bloccato in un certo punto del programma: ma l'errore potrebbe anche essere un poco prima o un poco dopo. Il programmatore deve riprendere in mano il suo programma sorgente e andare a ricontrollare il punto indicato e i suoi dintorni.
Identificato e corretto l'errore, la nuova versione del programma sorgente viene sottoposta a un nuovo procedimento di compilazione: se c'è ancora qualche errore, la compilazione si interrompe nuovamente e il programmatore deve nuovamente intervenire sul programma sorgente. E via di questo passo fino a che tutti gli errori sono stati eliminati.
Il processo è lento: il compilatore ha molto lavoro da svolgere, per tradurre il programma – tanto piú quando il programma è di grandi dimensioni, magari di qualche migliaio di istruzioni come succede tranquillamente con i programmi commerciali.
Bisogna ricordare comunque che, quando la traduzione (interpretazione o compilazione che sia) è completa, ciò non significa automaticamente che il programma sia anche perfetto! Il compilatore e l'interprete si basano esclusivamente sulle regole grammaticali del linguaggio, ma non possono stabilire la sensatezza o meno di quello che si trovano davanti. Se il programma contiene un errore fattuale o un errore di logica il compilatore non se ne può accorgere.

Quale linguaggio?

Si cita spesso un detto di Carlo V: "Parlo in spagnolo con Dio, in italiano alle donne e in tedesco al mio cavallo". Si può adattare il detto anche ai linguaggi di programmazione: non ne esiste uno assolutamente migliore degli altri, tutti hanno le loro particolarità, le loro idiosincrasie, i loro punti forti e i loro punti deboli. Ogni linguaggio di programmazione porta i segni degli obiettivi primari per i quali è nato, del tipo di problema che doveva risolvere chi l'ha formulato. In genere si può riuscire a fare quasi tutto con qualsiasi linguaggio di programmazione non troppo specializzato: proprio come si può dire in tedesco tutto quello che si dice in italiano. È vero però che certe cose si dicono meglio, o almeno piú efficacemente in una lingua piuttosto che in un'altra: Mozart preferiva, per le sue opere, libretti in italiano; per una moderna canzone "rock" è piú facile scrivere un testo in inglese che in italiano. Gli eschimesi, per i quali la neve è quasi tutto, non hanno un solo vocabolo per "neve", ma una ventina: non conoscono la neve generica, ma una ventina di tipi diversi di neve. Qualcosa di analogo succede con i linguaggi di programmazione: il COBOL, per esempio, è nato esplicitamente per l'elaborazione di dati commerciali e per tutti i programmi di tipo gestionale rimane un linguaggio eccellente: non ha funzioni matematiche complesse, ma in compenso strumenti agili per la gestione di

dati e la realizzazione di rapporti; APL è un linguaggio molto orientato alla matematica, e permette di lavorare con le matrici con la stessa semplicità con cui in BASIC o in COBOL si lavora con i numeri interi. Manipolare in APL stringhe di caratteri è possibile ma abbastanza complicato; SNOBOL4 e LISP, invece, sono nati proprio per l'elaborazione di stringhe e di liste. Le prime versioni del LISP erano debolissime per quanto riguarda le operazioni aritmetiche (poi questo linguaggio è stato ampliato e rafforzato in questo settore), FORTRAN invece è nato proprio per l'elaborazione scientifica. E via di questo passo.

Esiste poi una miriade di linguaggi molto "dedicati", pensati cioè per applicazioni specifiche: linguaggi per la grafica, per la musica, per il controllo industriale, per la simulazione.

Questi sono linguaggi specializzati: con un linguaggio studiato espressamente per la composizione musicale non si possono anche costruire programmi per il controllo di un robot.

A questo punto, la differenza fra un linguaggio di alto livello e un programma applicativo può diventare sfumata: taluni programmi di tabellone elettronico, per esempio, sono tanto ricchi da poter essere classificati effettivamente come linguaggi relativamente specializzati. Tanto per fare un esempio con un programma che ha avuto un grande successo: è di questo tipo 123 prodotto dalla Lotus Corporation. In 123 si scrivono effettivamente dei programmi che permettono di fissare una volta per tutte serie anche molto complesse di operazioni. La maggior parte dei pacchetti di software integrato si comportano allo stesso modo, e alcuni dichiarano esplicitamente la presenza di un "linguaggio" di programmazione specializzato.

Quale linguaggio di programmazione si deve scegliere, allora? Non esiste una ricetta valida per tutti: bisogna analizzare le proprie esigenze e documentarsi. I personal computer vengono forniti quasi tutti corredati del BASIC: è un buon punto di partenza, anche se non necessariamente il migliore. Il BASIC è fondamentalmente un linguaggio semplice, pensato all'origine per insegnare le tecniche elementari della programmazione a persone del tutto digiune dell'argomento. In seguito è stato progressivamente arricchito, e oggi è un linguaggio semplice ma potente e, per cominciare, ha anche il pregio di essere normalmente interpretato, quindi di permettere l'interattività.

Programmare è progettare

Questo è uno "slogan" davvero efficace: scrivere un programma significa stendere un progetto, un progetto di lavoro per il computer. Per la maggior parte, i linguaggi di programmazione sono di tipo *imperativo*: i programmi sono costituiti da ordini impartiti alla macchina. Scrivere un programma per il computer non è diverso dall'impartire una serie di ordini a una persona per farle eseguire un compito che non ha mai svolto in precedenza: solamente, il computer è piú "stupido" e ha bisogno di essere "guidato per mano" molto piú in dettaglio.

Posso benissimo dire a Giovanni: "prendi un uovo" e basta. Non devo anche suggerirgli che, se l'uovo non è a portata di mano, deve alzarsi e cercarlo nel frigorifero. Nel caso di un computer, il livello di dettaglio a cui si deve scendere è invece di questo tipo: può essere necessario addirittura spiegare che cos'è un frigorifero, che cos'è un uovo e quali muscoli bisogna mettere in azione per alzarsi e muoversi.

Fino a quale livello di dettaglio ci si deve spingere? Dipende da linguaggio a linguaggio. Il BASIC, come la maggior parte dei linguaggi di programmazione, ha un'istruzione specifica per la moltiplicazione fra interi, che viene indicata con il simbolo *. In BASIC, dire "233 * 84" è sufficiente per far capire alla macchina che si vuol moltiplicare fra loro i numeri 233 e 84. Il FORTH, invece, per esempio, non ha un'istruzione specifica per la moltiplicazione: allora, per dire alla macchina in FORTH che deve moltiplicare i due numeri 233 e 84, bisogna in realtà darle le istruzioni per simulare la moltiplicazione con i mezzi a disposizio-

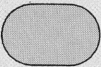
ne: per esempio, che deve predisporre una variabile di controllo da far variare fra 0 e 84, poi deve cominciare a sommare il numero 233 a se stesso per varie volte; ogni volta deve incrementare di un'unità la variabile di controllo e verificare se il suo valore è minore di 84; se lo è, deve continuare a sommare; quando il valore della variabile arriva a 84, il procedimento è finito e la somma ottenuta a quel punto è il risultato richiesto della moltiplicazione di 233 per 84. Ben diverso, vero? In realtà è piú semplice scrivere un programmino che spieghi una volta per tutte alla macchina che cosa deve fare quando incontra un simbolo * messo fra due numeri, e poi richiamare il programmino tutte le volte che serve. Questa è una caratteristica importante per un linguaggio di programmazione: la possibilità di definire funzioni e sottoprogrammi. I vari linguaggi non si equivalgono sotto questo profilo: alcuni offrono maggiori possibilità di altri. Alcuni linguaggi sono progettati con dovizia di funzioni già incorporate all'origine, e pochi strumenti per allargare il repertorio; altri invece sono piú agili, offrono solo un nucleo essenziale di funzioni già incorporate, ma offrono al programmatore strumenti molto potenti per crearsi su misura nuove funzioni. In alcuni linguaggi è facile creare "biblioteche" di programmi che possono essere incorporati in altri programmi piú complessi semplicemente mediante una "chiamata"; in altri linguaggi questa operazione è difficile o magari addirittura impossibile.

Il FORTRAN è particolarmente famoso per la biblioteca di funzioni e di sottoprogrammi (*routine*, in gergo) che è stata costruita nei molti anni trascorsi dalla sua prima formulazione. Moltissime funzioni non previste all'origine nella definizione di questo linguaggio sono state programmate in seguito e archiviate, divenute magari di pubblico dominio: uno scienziato adotta spesso il FORTRAN proprio perché non deve preoccuparsi di programmare molte funzioni specifiche, che trova già raccolte nella biblioteca di sistema (è il motivo fondamentale della longevità del FORTRAN, linguaggio nato ancora negli anni Cinquanta).

Il livello di dettaglio a cui si deve arrivare nella programmazione è dunque quello reso necessario dal linguaggio in cui si programma e dagli strumenti che mette a disposizione. Il progetto del programma, però, nelle sue linee essenziali può essere indipendente dal linguaggio di programmazione: è possibile, al-

Una sorta di albero-mappa dei linguaggi, riportato da Rich Didday nel secondo volume, dedicato al software, della sua Intervista sul personal computer *(Muzzio, Padova, 1981). In alto, al di là della "grande barriera" dell'intelligenza, stanno i linguaggi umani sia artificiali (esperanto, matematica), sia naturali. A fronte, una piccola cronologia con le date di pubblicazione dei principali linguaggi di alto livello per la programmazione dei calcolatori.*

meno entro certi limiti, la definizione di progetti – *algoritmi*, per usare il termine tecnicamente piú appropriato – in modo indipendente.
Che cos'è un algoritmo? È la formulazione esplicita del metodo di risoluzione di un problema, la specificazione del progetto da eseguire. Scrivere un programma per risolvere il problema *P* vuol dire tradurre in un linguaggio di pro-

grammazione specifico un algoritmo che dia, passo per passo, il metodo di risoluzione del problema, cioè la serie delle attività che conduce a risolverlo. Un algoritmo è piú o meno come una ricetta. Volete preparare una torta di mele? Allora prendete questi ingredienti, combinateli in questo modo, mettete in forno per tanto tempo alla temperatura tal dei tali, e via dicendo.
Non è detto che per risolvere un problema esista un metodo unico: algoritmi diversi possono portare allo stesso traguardo. In pratica non esiste mai un modo unico per costruire un programma che svolga un certo compito: però alcuni metodi (alcuni algoritmi) risultano migliori di altri, perché piú brevi, piú veloci, piú efficienti. Ma attenzione: quel che magari è piú breve o piú efficiente per noi può non esserlo per il computer o per il linguaggio di programmazione che usiamo.
Ricordate il massimo comun denominatore di due numeri? (Se ne parla abbondantemente nelle scuole medie.) Lo si abbrevia di solito con la sigla MCD, e lo si definisce come il numero piú grande che divida esattamente i due numeri dati. 4, per esempio, è il MCD fra 12 e 20.
Come si calcola il massimo comun divisore di due numeri? Il metodo insegnato tradizionalmente a scuola dice di scomporre in fattori primi i due numeri, poi di identificare i fattori comuni e di moltiplicarli per ottenere il massimo numero che divida entrambi i numeri di partenza. Il procedimento è abbastanza intuitivo per un essere umano, e funziona anche abbastanza bene, almeno per numeri non troppo grandi, ma richiede un buon numero di divisioni, eseguite mentalmente o sulla carta.
Questo metodo per un computer è inutilmente complicato: lunghe serie di divisioni (la divisione non è mai presente come operazione elementare, a livello di set di istruzioni del microprocessore), molti confronti, ancora operazioni relativamente complesse come le moltiplicazioni alla fine. Esiste però un altro algoritmo, meno intuitivo per noi, ma estremamente piú efficiente per la macchina, che non chiama in questione il concetto di numero primo e non richiede di fattorizzare (cioè di scomporre in fattori primi) i due numeri di partenza: è l'algoritmo di Euclide e, come il nome vi avrà fatto sospettare, non è una scoperta dell'ultima ora, ma un risultato noto ai matematici da oltre 2000 anni. L'algoritmo di Euclide può essere organizzato in tre passi:

1. Si divide il maggiore dei due numeri (chiamiamolo *a*) per il minore (chiamiamolo *b*) e si considera quindi il resto *r* della divisione.
2. Se *r* è 0, allora *b* è il MCD fra *a* e *b* e l'algoritmo si conclude.
3. Se invece *r* è diverso da zero, si torna al passo 1, mettendo *b* al posto di *a*, *r* al posto di *b*.

Il passo 3 è caratteristico dei meccanismi di programmazione: istituisce un *ciclo* o *loop*, secondo la terminologia anglosassone, che viene ripetuto fino a che non si ottiene una divisione con resto zero. Il divisore di quell'ultima operazione è il MCD cercato.
Tutto l'algoritmo si regge sul riconoscimento di una proprietà che può non venire subito alla mente: se un numero divide esattamente tanto *a* quanto *b* (e tale deve essere per definizione il loro MCD), allora quel numero deve dividere anche *a-qb*, dove *q* è il quoziente della loro divisione: e *a-qb* altro non è se non il resto della loro divisione! Quindi il MCD tra due numeri deve essere anche MCD fra il minore dei due e il resto della loro divisione. Se non siete perfettamente convinti della cosa, potete provare a calcolare il MCD per un po' di coppie di numeri, con il metodo di Euclide, e verificare che funziona davvero.
L'algoritmo di Euclide non è affatto immediato per noi (per lo meno, se non siamo matematici): ci viene piú spontaneo andare alla ricerca dei fattori dei due numeri. Non c'è dubbio, invece, che per la macchina (per la quale nulla è intuitivo) il procedimento euclideo sia nettamente piú semplice ed efficiente: richiede solo una serie relativamente breve di divisioni, che è in grado di svolgere con grande rapidità.

I 4 simboli fondamentali utilizzati nella preparazione dei diagrammi di flusso.

Rappresentazione di algoritmi

Gli esperti di programmazione hanno sviluppato dei metodi per rendere piú facile seguire mentalmente il meccanismo di funzionamento di un algoritmo, comunicarlo ad altri e identificare la struttura generale di un programma che realizzi in un linguaggio particolare l'algoritmo stesso. Il piú noto e il piú semplice di questi metodi è il cosiddetto *diagramma di flusso* o *diagramma a blocchi* (*flow chart* in inglese).
In un diagramma di flusso i singoli passi di un algoritmo sono indicati all'interno di "blocchi", simboli grafici la cui forma indica la natura del passo stesso, uniti da linee che indicano il percorso (o i percorsi) del procedimento. A pagina 154 si può vedere il diagramma di flusso per l'algoritmo di Euclide del massimo comun divisore. Gli ovali indicano l'inizio e la fine del procedimento; i rettangoli contengono la descrizione di operazioni; i rombi (che per comodità di scrittura possono essere anche esagoni allungati) contengono decisioni, punti cioè in cui chi segue l'algoritmo deve eseguire qualche tipo di controllo e, in base al risultato del controllo, deve prendere una strada piuttosto che un'altra. I rombi indicano decisioni e sono punti in cui il percorso dell'algoritmo si dirama: nel nostro caso, il rombo indica il momento in cui si deve andare a guardare il resto e, se è 0, si deve considerare concluso il procedimento, mentre, se è diverso da 0, si deve tornare indietro ed eseguire nuovamente l'operazione di divisione, dopo aver modificato opportunamente i due numeri su cui operare.
Dal diagramma si può capire bene perché sia stato dato il nome di *ciclo* all'iterazione che si instaura in questo modo: il percorso dell'algoritmo si chiude circolarmente su se stesso.
In un diagramma di flusso in cui non compaiono rombi ma solo rettangoli e gli ovali di inizio e fine, l'algoritmo ha solo un andamento lineare: i vari passi vengono eseguiti in sequenza, senza mai ritorni e senza salti.
Il ciclo e l'iterazione sono due fra le *strutture di controllo* fondamentali in qualunque programmazione: la terza struttura fondamentale è l'alternativa: "se si verifica una certa condizione fai la tal cosa, altrimenti fai la talaltra", che in un diagramma di flusso visivamente si presenta come una suddivisione dell'albero in due rami, i due percorsi da prendersi nei due casi identificati.

Il diagramma di flusso non è l'unico modo in cui si può rappresentare un algoritmo, ma è certo, soprattutto agli inizi, il piú perspicuo ed efficace. Per chi è piú esperto, può essere piú rapida la rappresentazione lineare in una sorta di "pseudocodice", un linguaggio regolato che si avvicina a un linguaggio di programmazione e ne ricalca le movenze.
Riprendiamo il nostro algoritmo euclideo e, intanto, aggiungiamo qualche piccolo accorgimento al nostro procedimento. Tanto per cominciare, notiamo che il quoziente non ci serve, in effetti: ci interessa conoscere solo il resto. Esiste un'operazione che identifica il resto, senza passare attraverso il quoziente: è

l'operazione modulo; "a modulo b", piú sinteticamente "a mod b" dà come risultato il resto della divisione di a per b: quasi tutti i linguaggi di programmazione mettono a disposizione questo operatore. Inoltre, aggiungiamo alla fine dell'algoritmo un'istruzione che ci visualizzi in uscita il massimo comun denominatore. Una notazione: tutti i linguaggi di programmazione usano variabili, analoghe a quelle dell'algebra, il cui ruolo è quello di identificare in modo comodo per l'utente una locazione (o un gruppo di locazioni) di memoria: ogni variabile è identificata da un nome, per la cui costruzione ciascun linguaggio fissa regole specifiche. A una variabile si assegna un valore (che può variare, per l'appunto): significa depositare un certo valore in una locazione di memoria (il vantaggio dell'uso di variabili è che non ci si deve preoccupare di quale sia esattamente questa locazione: la si usa mediante il suo nome). L'operazione di assegnazione viene indicata, nella stesura di algoritmi, quasi sempre con una freccia: x ← 24 significa "assegnamo il valore 24 alla variabile x".
Ecco, con queste convenzioni, un nuovo modo di rappresentare l'algoritmo di Euclide:

x è una variabile
y è una variabile
z è una variabile

INIZIO

    x ← a; y ← b

ripeti z ← x mod y

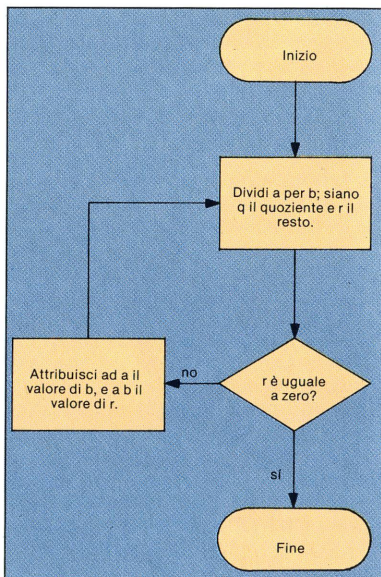
    x ← y
    y ← z

finché z = 0

stampa "il MCD di a e b è" y

FINE

Un po' diverso da prima? Non molto, è solo tutto un pochino piú esplicito. Cominciamo con il dichiarare che *x*, *y* e *z* sono delle variabili (non tutti i linguaggi di programmazione lo richiedono, a dire il vero, ma può essere utile segnalarlo). Poi mettiamo la parola INIZIO per identificare il punto in cui comincia l'algoritmo vero e proprio (alcuni linguaggi hanno effettivamente un costrutto *begin... end*); segnaleremo la conclusione con la parola FINE. Il primo passo è poi quello di assegnare alle variabili *x* e *y* i valori dei due numeri di cui si cerca il MCD; poi procediamo a calcolarne il resto (*x mod y*) e assegnamo tale valore a *z*. Poi ci prepariamo a ricominciare da capo, assegnando il valore di *y* a *x* e il valore di *z* a *y*: se il resto è diverso da zero, dovremo procedere a dividere il numero *b* per il resto, e con il sistema che abbiamo adottato non dobbiamo dare una nuova istruzione, ma ci limitiamo a riassegnare i valori alle variabili in modo da continuare a utilizzare sempre l'istruzione *x mod y* come fulcro del nostro algoritmo. In questo modo, con tre sole variabili possiamo procedere per

Il diagramma di flusso per l'algoritmo di Euclide del massimo comun divisore.

un numero qualsiasi di ripetizioni del ciclo. Prima di riprendere il ciclo, però, la clausola *finché z = 0* ci invita a controllare il valore di z (che, si badi bene, non è stato toccato dalle istruzioni di assegnazione precedenti): se il valore è diverso da 0, si riprende il ciclo (abbiamo detto: *ripeti... finché*), se invece il valore è 0, si esce dal ciclo e si passa all'istruzione successiva, che dice di stampare la frase fra virgolette cosí com'è e di aggiungere alla fine il valore della variabile z.
Il *ripeti... finché* è uno dei modi in cui si può concretizzare la struttura di controllo dell'iterazione. Un altro modo fondamentale è il *mentre... ripeti*: la differenza sta nel fatto che nel *ripeti... finché* il controllo avviene alla fine di ogni ripetizione del ciclo, nel *mentre... ripeti* il controllo avviene all'inizio: nel primo caso le istruzioni del ciclo vengono eseguite almeno una volta, mentre nel secondo caso possono anche non venire eseguite, se la condizione non è soddisfatta già al primo controllo (prima dell'ingresso nel ciclo).
In alcuni linguaggi di programmazione queste due strutture di controllo sono presenti proprio in questo modo: nel PASCAL, per esempio, esistono i costrutti *repeat ... until* e *while ... do*; non tutti i "dialetti" del BASIC, invece, le possiedono. In quelle versioni del BASIC che non possiedono questi costrutti, è possibile simularli in altro modo.
La sequenza, l'iterazione e l'alternativa sono le tre strutture di controllo fondamentali: si dimostra infatti che qualunque algoritmo può essere rappresentato facendo uso di queste tre sole strutture.

Un po' di BASIC

Il BASIC è il principale linguaggio di programmazione per i personal computer: è nato nella prima metà degli anni Sessanta, opera di John G. Kemeny e Thomas Kurtz, al Dartmouth College nel New Hampshire. Kemeny e Kurtz volevano un linguaggio relativamente semplice, attraverso il quale accostare gli studenti alla programmazione: BASIC è l'acronimo di *Beginner's All-purpose Symbolic Instruction Code*, ovvero "codice d'istruzioni simbolico di uso generale per principianti". Negli anni Sessanta i personal computer non esistevano ancora, e le prime applicazioni del BASIC avvennero sui grandi sistemi in *time-sharing* (ovvero a partizione di tempo, denominazione con cui si indicava in sintesi il modo di utilizzazione dell'unità centrale da parte di utenti diversi). Probabilmente non sarebbe sopravvissuto fino a oggi, e sicuramente non avrebbe avuto la diffusione che ha, se non fosse stato proprio per la nascita dei sistemi di elaborazione personali: sistemi rivolti a utenti non professionisti dell'informatica, per i quali erano necessari strumenti di programmazione semplici ma efficaci. Il BASIC poi era un linguaggio interpretato, che permetteva l'interattività e quindi anche la sperimentazione in un dialogo immediato con la macchina: non era stato pensato proprio per i principianti?
Abbiamo già visto nelle prime pagine come sia possibile usare il BASIC in modo immediato, semplicemente battendo alla tastiera dei comandi, a cui la macchina, per il tramite dell'interprete, risponde immediatamente. Abbiamo anche visto qualche esempio elementare di uso in modo differito, cioè mediante programma: è sufficiente la numerazione delle righe perché l'interprete "veda" ciò che scriviamo come una serie di istruzioni da eseguire collettivamente come un unico programma, e solo dietro nostra esplicita richiesta (il comando – dato in modo immediato – RUN). Non è certo questa la sede piú adatta per una trattazione esaustiva del BASIC: per questo esistono libri appositi, ben piú voluminosi, esplicitamente pensati con intenti didattici, e i manuali forniti dalle singole case costruttrici. E, inoltre, non ci sembrerebbe corretto: non è affatto detto, come abbiamo già sottolineato, che il BASIC sia per forza il linguaggio piú adatto. Oggi la comunità degli appassionati del personal computer è largamente cresciuta, cosí come sono cresciuti gli strumenti disponibili: se il BASIC resta il linguaggio piú diffuso, esistono interpreti e compilatori anche per

molti altri linguaggi (cosa che certo non si poteva dire solo pochi anni addietro) e si può operare una scelta sulla base di qualunque tipo di motivazione. Oggi ci si può accostare al mondo del personal computer anche perché si è attratti dai robot programmabili, dalla grafica o dalla musica, o ancora dal fascino delle ricerche sull'intelligenza artificiale: e per nessuno di questi campi il BASIC è il migliore linguaggio disponibile.

In omaggio alla sua diffusione, però, è giusto dedicare un po' piú di spazio a questo linguaggio: non imparerete a programmare in BASIC in queste pagine, ma potrete averne un'idea un po' piú precisa.

Abbiamo visto l'istruzione PRINT: PRINT è una delle "parole riservate" del linguaggio. Le parole riservate sono parole con un significato prestabilito dal linguaggio: possono essere usate solo con quel significato. Non è possibile cioè usare PRINT come nome di variabile: l'interprete la vede sempre come un'istruzione, e darà risultati sbagliati o segnalerà errori di sintassi. Come PRINT, esistono numerose altre parole riservate: molte sono comuni a tutte le versioni del BASIC, altre sono proprie di "dialetti" particolari.

Sono "parole riservate" del BASIC i simboli per le operazioni aritmetiche fondamentali: + – * / ^ (somma, differenza, prodotto, divisione ed esponenziazione). Il BASIC permette di lavorare su numeri interi, con una precisione limitata: fra +32767 e –32768, per esempio. Il problema della limitazione è sempre elevato, con qualsiasi macchina. Si può lavorare anche con numeri "reali", cioè con numeri decimali, in cui compare il punto decimale (la notazione usata sui calcolatori è sempre quella anglosassone, con il punto al posto della virgola decimale, e senza separazioni fra le centinaia e le migliaia, e via dicendo). Per i numeri reali è usata spesso la notazione cosiddetta *scientifica* o *esponenziale*, che rappresenta il numero con una *mantissa* (eventualmente con segno) e un esponente, preceduto dal segno E: 2.035E6 significa 2.035 × 10^6, cioè 2035000.

La precisione è in genere di sei cifre, ma su alcune macchine si possono avere costanti numeriche in precisione doppia, con 16 cifre. La limitazione può essere sempre superata con un programma: bisogna insegnare alla macchina a effettuare i calcoli con un algoritmo come quelli che applichiamo noi (non è banale, comunque con programmi opportuni si può ottenere qualunque precisione, almeno in teoria).

Le variabili numeriche si identificano con un suffisso: % per le variabili intere, ! (o nessun suffisso) per le variabili in precisione singola, # per quelle in precisione doppia. In mancanza di indicazioni, una variabile è intesa in precisione singola:

MCD% è una variabile intera

LIMITE è una variabile in precisione singola

PUNTO # è una variabile in precisione doppia.

Non è necessario dichiarare le variabili all'inizio dei programmi (come invece richiede, per esempio, il PASCAL): il tipo della variabile è riconosciuto dalla forma del suo nome.

Abbiamo visto che si possono elaborare non solo numeri, ma anche caratteri: il suffisso $ indica una *variabile di stringa*, i cui valori possono essere solo stringhe, cioè successioni, di caratteri.

Si assegna un valore a una variabile con il simbolo =:

A$= "QUESTA È UNA VARIABILE DI STRINGA"

è una istruzione di assegnazione, in cui alla variabile (di stringa, come si deduce dal segno di dollaro) A$ viene assegnato come valore la stringa di caratteri QUESTA È UNA VARIABILE DI STRINGA. È da notare come, nel caso di una variabile di stringa, la successione di caratteri assegnata come valore debba essere compresa fra virgolette.

È possibile avere anche l'equivalente di vettori e matrici, di cui si possono fissare le dimensioni con una apposita istruzione DIM: per esempio, DIM B%(5) definisce un vettore a cinque componenti, ciascuna delle quali è un numero intero; DIM A$ (4, 7) definisce una matrice a due dimensioni, con quattro righe di set-

te colonne ciascuna, i cui elementi sono stringhe di caratteri. Una matrice come questa, per esempio, potrebbe essere utilizzata per memorizzare i cognomi degli ospiti di un albergo: ogni riga corrisponde a un piano, ogni colonna a una delle sette camere poste su ciascun piano. Il nome del cliente ospitato al terzo piano nella stanza 5 sarebbe identificato dall'elemento A$ (3, 5). Una matrice (un vettore è in fondo una matrice unidimensionale) è una comoda *struttura di dati* presente in quasi tutti i linguaggi di programmazione. Nel BASIC la si trova indicata anche con il nome di "variabile con indici".
Sulle variabili di stringa si possono effettuare anche operazioni: il + indica la concatenazione di stringhe, LEN dà la lunghezza di una stringa, MID$ (x, y) dà gli y caratteri centrali di una stringa a partire dall'x-esimo; LEFT$ (x) e RIGHT$ (x) danno i primi x caratteri a sinistra e a destra, rispettivamente, di una stringa, e cosí via. Esistono anche due funzioni che permettono di trasformare numeri in caratteri e viceversa: sono le funzioni CHR$ e ASC, la prima delle quali dà il codice ASCII (in decimale) corrispondente a un carattere, mentre la seconda esegue l'operazione inversa, fornisce cioè il carattere corrispondente a un numero decimale interpretato come codice ASCII.
L'uguale (=) che abbiamo visto usato come operatore di assegnazione di valore a una variabile, viene usato in altri contesti anche come simbolo di relazione (cioè come simbolo di uguaglianza in senso stretto): come = vengono usati anche i simboli > (maggiore), < (minore), < = (minore o uguale), > = (maggiore o uguale), < > (diverso). Questi operatori di relazione sono usati in particolare per definire le condizioni di controllo, all'interno di strutture come l'alternativa e l'iterazione.
Questo ci porta alle strutture di controllo del BASIC. L'alternativa è data dal costrutto IF ... THEN (in qualche versione con la clausola ELSE). Per esempio:

```
100 IF X = Y THEN PRINT "CIAO"
110 PRINT "BUONGIORNO"
```

è un'alternativa dove la condizione è data da X = Y: se X è uguale a Y si passa a eseguire quanto indicato dopo il THEN (in questo caso, stampare CIAO); in caso contrario si passa subito alla riga successiva (e qui si stampa BUONGIORNO). L'esempio è un po' stupido, ma quel che importa è il meccanismo.
La forma tipica dell'iterazione usa il costrutto FOR, con una variabile di controllo che può assumere un certo insieme prefissato di valori:

```
100 FOR I = 1 TO 10
110 PRINT 5 * I
120 NEXT I
```

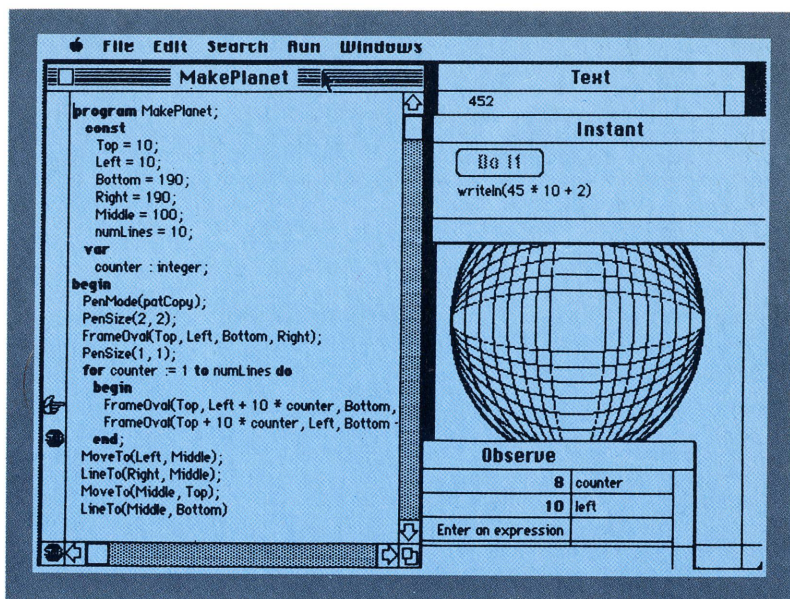
Queste tre istruzioni dicono: usiamo I come variabile di controllo, che può variare fra 1 e 10 (si sottintende, con incrementi di una unità) e a ogni passaggio stampiamo il valore che si ottiene moltiplicando 5 per il valore che ha in quel momento I stessa. Il risultato è la successione dei primi dieci multipli di 5: 5, 10, 15, 20, ecc.
La I è la *variabile di controllo*: l'espressione I = 1 TO 10 fissa il campo di variazione; si può aggiungere la parola STEP seguita da un numero per modificare l'incremento a ogni passaggio. FOR I = 0 TO 100 STEP 10, per esempio, dice che la variabile I deve variare fra 0 e 100, con incrementi di 10 unità. (STEP, che letteralmente significa "passo", può assumere anche valori negativi, dando cosí dei decrementi alla variabile di controllo).
La clausola NEXT I segnala la chiusura del ciclo e il ritorno all'inizio della serie di operazioni da ripetere: le istruzioni del ciclo sono tutte quelle racchiuse fra il FOR e il NEXT.
È da notare che i cicli FOR ... NEXT possono anche essere "nidificati" uno all'interno dell'altro, cioè è possibile avere cicli che si sviluppano all'interno di altri cicli.
Abbiamo visto l'alternativa e l'iterazione: la sequenza è la modalità normale, ed è assicurata dalla numerazione delle righe. L'interprete, cioè, provvede a leggere ed eseguire le istruzioni in successione, secondo l'ordine crescente dei

numeri di riga, a meno che non incontri istruzioni che modifichino questo andamento, attribuendo il controllo della macchina a istruzioni diverse da quelle che seguono, nella successione normale. Fra queste istruzioni di controllo, oltre l'alternativa IF ... THEN e l'iterazione FOR ... NEXT, il BASIC possiede anche un'istruzione di salto GOTO (letteralmente "vai a"). Il GOTO seguito da un numero di riga passa il controllo alla riga con quel numero: da lí l'esecuzione procede poi normalmente.

Il salto può essere *condizionato* o *incondizionato*: il GOTO, cioè, può essere autorizzato dal verificarsi di una condizione, oppure essere richiesto "in assoluto". Nel primo caso si ha una situazione del tipo "se le cose stanno in questo modo, vai (GOTO) alla riga xxx": questo tipo di meccanismo è necessario per simulare strutture di controllo come il "ripeti... finché", in tutte le versioni del BASIC che non presentano già questo costrutto. In questo caso è necessario impostare la successione di operazioni da iterare; al termine chiedere la verifica della condizione con l'IF e, in caso, rimandare alla prima istruzione della serie con il GOTO.

Il GOTO incondizionato è stato la pietra dello scandalo, negli anni Settanta, per tutti coloro che sostenevano la necessità di un modo di programmare ordinato, "strutturato", chiaro e sottoposto a regole precise. Il salto incondizionato, in effetti, altera pesantemente la successione regolare delle istruzioni, permettendo di correre avanti e indietro lungo il programma. In qualche caso è molto comodo, ma unendo questa caratteristica al fatto che il BASIC, essendo un interprete, incoraggia una programmazione per accumuli successivi, a partire da un nucleo a cui si aggiungono a piacere pezzi ulteriori, nel colloquio interattivo con la macchina, si può ottenere quella che pittorescamente gli americani hanno soprannominato "spaghetti programming", programmazione a spaghetti. Ovvero, una programmazione intrecciata e caotica, che permette di creare con facilità, ma rende difficile la lettura dei programmi a chi non li ha scritti in prima persona, rende difficile al creatore stesso l'identificazione delle proprie idee, se per caso si trova a riprendere in mano la propria creatura a distanza di tempo. Evitare il GOTO incondizionato è dunque meglio.

Di ben altra natura è il GOSUB, che permette il passaggio da un punto del programma a un altro punto, ma costringe l'interprete a tener conto del punto di partenza e a ritornarvi, non appena incontra l'istruzione gemella RETURN. Esso consente di creare dei sottoprogrammi separati dal corpo principale del programma: cosa utile quando, all'interno di un programma, occorre eseguire varie volte, ma in punti diversi, uno stesso procedimento (una stessa *routine* o, piú esplicitamente, un particolare algoritmo). In questo caso non occorre ripetere la serie di istruzioni che realizza l'algoritmo nei vari punti del programma in cui è richiesta: è sufficiente scriverla una volta sola, alla fine del programma principale, in modo che la numerazione delle righe non crei pasticci, quindi si può richiamarla con l'istruzione GOSUB ogni volta che serve.

Il BASIC mette a disposizione uno strumento molto semplice per *documentare* i programmi: le istruzioni REM (abbreviazione per *remark*, che significa "nota, osservazione"), che non sono a dire il vero vere e proprie istruzioni. Quando l'interprete incontra la parola chiave REM trascura tutto quello che segue su quella riga: dopo un REM il programmatore può inserire allora annotazioni che chiariscono il significato delle istruzioni che seguono. Un buon uso delle REM è sempre raccomandabile. La facilità con cui un programma in BASIC può essere corretto, ampliato con l'inserimento di ulteriori righe, modificato localmente, incoraggia in effetti un modo di procedere per piccoli passi, per tentativi ed errori, che sicuramente rende molto agevole il primo confronto con una macchina, ma a lungo andare, se non è opportunamente disciplinato, favorisce la frammentazione e il disordine. È un po' come con le costruzioni dei bambini: si parte con l'idea di fare una casetta, poi si aggiunge un cubetto qua,

Il PASCAL per il Macintosh è interpretato, non compilato: qui lo vediamo all'opera, nell'ambiente a finestre di questa macchina.

un cubetto là, e alla fine si ottiene un mulino a vento o una nave. Il procedimento è splendido per l'esplorazione casuale, ma a lungo andare è controproducente per la programmazione: si riesce a far fare qualcosa alla macchina, ma dopo un po' non si riesce piú a capire come si è riusciti a farglielo fare. Documentare abbondantemente i propri programmi, in questi casi, è estremamente consigliabile.

Il BASIC incoraggia, dunque, una programmazione per piccoli passi, per tentativi ed errori: la incoraggia, non la rende però inevitabile. Purtroppo anche molti manuali di istruzione, rivolti ai neofiti, incoraggiano questo stile: è divertente, ma fondamentalmente errato. Dopo le prime sperimentazioni, è molto meglio ricordare che la programmazione è una forma di progettazione: e i programmi sono validi e chiari solo se le idee di chi li scrive sono valide e chiare.

È importante chiarirsi bene le idee prima di cominciare a stendere un programma. Un diagramma di flusso, una descrizione dettagliata dell'algoritmo in un qualsiasi pseudocodice lineare possono risparmiare tempo e fatica e rendere molto piú efficace il risultato finale.

Il BASIC non è stato pensato per una applicazione particolare: è un linguaggio *all-purpose*, di uso generale, come dice il suo nome per esteso. Questo significa che può essere utilizzato per scrivere programmi in ogni campo, ma anche che non dispone di risorse specifiche per compiti di particolare complessità. È un linguaggio adatto per affrontare genericamente la programmazione, per avvicinarsi al mondo dell'elaborazione, e per i personal computer è un linguaggio eccellente proprio perché non vincola l'utilizzazione a settori specifici. I programmi scritti in BASIC non sono per questo necessariamente banali: scrivere buoni programmi o programmi complessi non dipende tanto dal linguaggio quanto dall'abilità e dall'intelligenza del programmatore.

Una volta scritto il programma, verificato che funzioni esattamente come ci si aspettava, si potrà scoprire che la sua esecuzione è un po' lenta: è il difetto tipico di tutti i linguaggi interpreti. Esistono però anche BASIC compilati: si può pensare di scrivere un programma con un interprete, per verificarlo interattivamente, per poi compilarlo prima di eseguirlo effettivamente. Se un programma deve essere usato spesso e la velocità è un fattore importante, può valer la pena: purché si disponga di un interprete e di un compilatore BASIC compatibili, che hanno cioè le stesse istruzioni. E questo discorso ci porta ai dialetti.

I dialetti del BASIC

Il BASIC di Kemeny e Kurtz ha subito, nei suoi venti anni di vita, una considerevole evoluzione: è stato infatti progressivamente ampliato, sino a diventare un linguaggio assai potente, ma al tempo stesso ogni costruttore ha cominciato a realizzare versioni tagliate su misura per i propri prodotti. I dialetti del BASIC non si contano piú: come avviene per le lingue, anche se tutti i dialetti hanno un nucleo comune abbastanza sostanzioso, le differenze possono non essere secondarie.
Il BASIC è un linguaggio di alto livello e, come abbiamo detto, un grande pregio dei linguaggi di alto livello è la loro trasportabilità, cioè la possibilità di essere usati su macchine diverse, indipendentemente dall'hardware di volta in volta utilizzato. Vero, ma solo fino a un certo punto: ogni costruttore tende a personalizzare il BASIC della propria macchina, con l'obiettivo di sfruttarne al meglio le capacità (ma a volte anche con l'obiettivo non dichiarato di rendere piú difficile per l'utente il passaggio a macchine di altri costruttori, perché dovrebbe rinunciare ai propri programmi o sobbarcarsi lo sforzo di un adattamento, a volte assai gravoso). Talvolta le personalizzazioni sono inevitabili: una macchina che consenta di gestire un video a colori deve disporre di un linguaggio in cui esistano istruzioni relative ai colori; le macchine orientate alla grafica hanno bisogno di istruzioni grafiche (e le caratteristiche grafiche variano da macchina a macchina); quelle che in piú dispongono di circuiti di generazione di suoni debbono fornire all'utente estensioni del linguaggio che permettano il controllo delle capacità sonore. Ed è ovvio che un programma per far eseguire una melodia al generatore di suoni non è trasportabile su una macchina che non disponga delle stesse capacità musicali. Da questo punto di vista, l'evoluzione dell'hardware degli elaboratori personali è ancora cosí lontana dall'esaurimento da rendere improponibile il congelamento del BASIC in una forma unica e immutabile, anche se qualche forma di standardizzazione sarebbe auspicabile. Uno standard di fatto è diventato, in certa misura, il BASIC della casa americana Microsoft, la quale tuttavia ha realizzato versioni un po' differenti del proprio interprete per macchine differenti. Sempre della Microsoft è il BASIC adottato da tutti i calcolatori che aderiscono allo standard MSX (vari costruttori giapponesi, la Philips in Europa), uno standard che investe sia l'hardware sia il software, pensato per rendere sostanzialmente intercambiabili le macchine che lo adottano.
Le differenze fra i dialetti del BASIC non si limitano, comunque, solo a differenze dovute alla grafica o ai suoni: si trovano differenze inoffensive, come parole chiave diverse per la medesima istruzione, e differenze sostanziali come la disponibilità o meno di talune istruzioni. Tranne che per programmi molto semplici, dunque, un programma scritto nel BASIC di un particolare sistema non girerà tale e quale su un sistema dotato di un altro dialetto del BASIC.

Gli altri linguaggi di programmazione

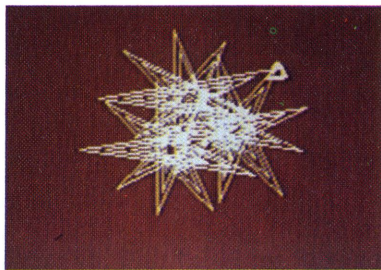
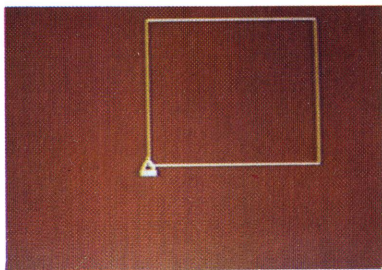
Esistono centinaia di linguaggi di programmazione diversi: solo alcuni hanno raggiunto una diffusione ampia e generalizzata. Molti ormai fra questi hanno raggiunto anche il mondo dei personal computer. Tra i linguaggi ancora vivi, il piú "anziano" è il FORTRAN, nato alla

metà degli anni Cinquanta: il nome è una sigla per "Formula Translator", traduttore di formule. Il FORTRAN è nato per le esigenze del mondo scientifico, nel quale è ancora largamente utilizzato: è un linguaggio compilato, che ha il pregio di consentire l'uso di sottoprogrammi esterni, compilati separatamente. In questo modo si possono creare biblioteche di sottoprogrammi pronti per l'uso, facilmente richiamabili: il che è estremamente vantaggioso nel mondo scientifico, dove difficilmente i procedimenti variano drasticamente e rapidamente. Nonostante l'età, in effetti, il FORTRAN continua a essere largamente usato proprio grazie all'enorme biblioteca di programmi che si è andata accumulando negli anni e che è diventata un patrimonio comune inestimabile. Esistono compilatori FORTRAN anche per i calcolatori personali: non presentano tutta la potenza del linguaggio, e le possibilità di calcolo dei personal sono limitate: tuttavia non si tratta di giocattoli, e possono essere usati con frutto in campo scientifico.
Decisamente piú giovane il PASCAL, nato alla fine degli anni Sessanta, opera di Niklaus Wirth, una delle figure chiave nella storia dell'informatica. Il PASCAL era, nelle intenzioni dell'autore, principalmente un linguaggio per lo studio delle buone tecniche di programmazione: è infatti un linguaggio ben strutturato, elegante. Wirth e i suoi collaboratori, tuttavia, hanno badato molto anche all'efficienza del loro compilatore, mirando a ottenere un programma traduttore che fornisse un codice oggetto il piú possibile compatto: per questo il PASCAL è risultato un linguaggio molto valido per il trasferimento sui piccoli calcolatori personali. È un linguaggio versatile, chiaro e potente, utilizzabile anche per applicazioni professionali: i compilatori per alcune macchine sono dotati anche di estensioni per il trattamento della grafica. È un linguaggio caldamente consigliabile a tutti coloro che abbiano un interesse non superficiale per la programmazione: conoscendo il PASCAL è abbastanza facile passare poi anche ad altri linguaggi, perché il PASCAL ha "fatto scuola" e molte idee analoghe si ritrovano in altri linguaggi piú recenti.
Dopo il PASCAL, Wirth ha sviluppato un altro linguaggio di programmazione, il MODULA-2, ancora piú potente, in cui la programmazione avviene per moduli indipendenti, che interagiscono fra loro mediante "funzioni" di "importazione" ed "esportazione". Il MODULA-2 è orientato alla costruzione di software cosiddetto "di sistema", cioè per i sistemi operativi e per i programmi di base di una macchina, quelli che servono alla gestione di tutte le sue risorse. Esistono alcuni compilatori MODULA-2 anche per alcuni personal computer, anche se si tratta di prodotti costosi.
Anche il C è un linguaggio nato per il software di sistema, ed è nato in un ambiente altamente professionale: i laboratori della Bell, negli Stati Uniti. C è un linguaggio essenziale: pochi costrutti fondamentali e potenti mezzi a disposizione del programmatore per crearsi i propri strumenti. È un linguaggio sintetico, poco adatto per i principianti, ma che in mano a programmatori già un po' esperti può dare risultati notevoli. Negli Stati Uniti, dall'inizio degli anni Ottanta, ha goduto di una diffusione crescente anche nel mondo dei personal: proprio perché è un linguaggio ridotto all'osso il suo compilatore può trovare posto anche nella memoria limitata di una macchina di piccole dimensioni.
Il FORTH è stato sviluppato nel corso degli anni Sessanta principalmente come linguaggio per il controllo di strumenti (per esempio, di grandi telescopi). Ha una impostazione di base analoga a quella del C: linguaggio fondamentale di piccole dimensioni, molto stringato, ma con mezzi potenti per la creazione di ulteriori strumenti, a discrezione del programmatore. Non è un linguaggio molto diffuso, ma per le sue caratteristiche può essere anche molto amato (o viceversa molto odiato). Eccellente per la programmazione di robot.
LISP è un altro linguaggio ricco di storia: la sua data di nascita risale alla fine degli anni Cinquanta. È anche il linguaggio principale degli studi sull'intelligen-

za artificiale. È molto diverso dai linguaggi di programmazione piú comuni, perché è stato orientato, sin dall'inizio, soprattutto al trattamento di liste (strutture di dati particolari, che sono quel che dice il nome, cioè successioni di elementi) di qualunque natura. LISP (*List processing* il nome per esteso, cioè "elaborazione di liste") è un linguaggio funzionale: tutto si fa con funzioni. È anche un linguaggio fortemente ricorsivo: la ricorsività è la sua caratteristica piú saliente. In altri linguaggi la ricorsività può essere una caratteristica fra le tante, magari neanche tanto facilmente maneggiabile; in LISP invece è proprio uno strumento principe. Un procedimento si dice ricorsivo quando richiama se stesso al proprio interno. L'esempio piú illuminante può essere dato dal modo in cui noi costruiamo le frasi nel linguaggio di tutti i giorni. Per esempio, una regola di sintassi dell'italiano dice che possiamo costruire una frase con soggetto e predicato verbale, dove il predicato verbale può essere un verbo seguito da complemento oggetto: ma il complemento oggetto può essere sostituito da un'altra frase, con una medesima struttura, al cui interno la struttura può ripetersi ancora: "Io credevo che tu pensassi che Giovanni dicesse..." è un tipico esempio di frase in cui la stessa struttura continua a ripetersi su se stessa. Un altro esempio è dato dalla definizione, in matematica, del fattoriale:
- il fattoriale di 0 è 1
- il fattoriale di n è n moltiplicato per il fattoriale di n – 1.

Questa è una definizione ricorsiva perché la funzione da definire è richiamata all'interno della definizione stessa. La definizione è comunque valida, perché c'è il primo passo che definisce in modo assoluto il fattoriale di 0: dopo la definizione procede, continuando a costruire su se stessa. Il procedimento di calcolo poi procede all'inverso: si vede come per calcolare il fattoriale di 5 si debba prima calcolare quello di 4, ma prima ancora quello di 3, prima ancora quello di 2 e prima ancora quello di 1: ma quest'ultimo è dato, e si può andare a chiudere tutti i passaggi. In BASIC, per esempio, non si può usare questa definizione del fattoriale: si deve procedere in un altro modo, perché il BASIC non permette la programmazione ricorsiva. Chiaramente un linguaggio come il LISP, che fa della programmazione ricorsiva il proprio cardine, deve avere un programma traduttore molto particolare, che deve tenere conto di tutti i livelli di ricorsività che debbono rimanere aperti fino a che non si raggiunge la base di un procedimento, e deve provvedere poi a chiuderli progressivamente, per dare il risultato desiderato. In questo modo produce moltissimo materiale intermedio, indispensabile ma non interessante per la risposta finale e, per esempio, deve essere dotato di procedure per la frequente esplorazione della memoria e la sua "ripulitura" dal materiale intermedio accumulato e non piú utile.

Il LISP è un linguaggio interpretato, come il BASIC: negli esperimenti di intelligenza artificiale, l'interattività con la macchina è un fattore importante. Esistono compilatori LISP anche per i personal computer: alcuni anche di notevole potenza.

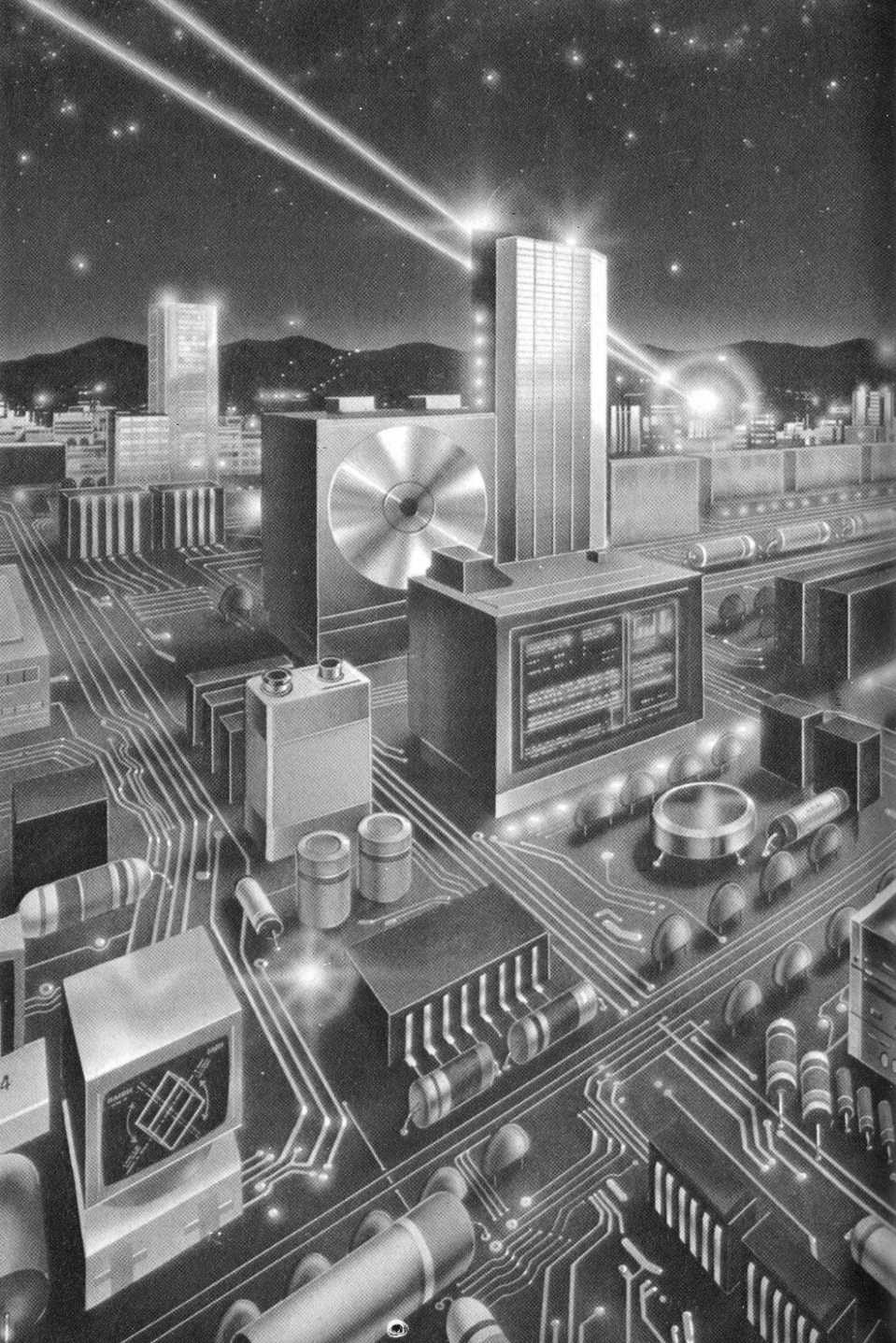
Come LISP, anche PROLOG è un linguaggio orientato all'intelligenza artificiale, e deve molta della sua fama al fatto che i ricercatori giapponesi lo abbiano eletto a linguaggio principale per il progetto di "calcolatori della quinta generazione", calcolatori per gli anni Novanta che dovrebbero possedere caratteristiche di "intelligenza" molto piú elevate di quelle delle macchine attuali. PROLOG ha una caratteristica notevole: non è un linguaggio algoritmico come quelli che abbiamo visto in precedenza. Non è richiesto, cioè, di spiegare alla macchina passo per passo come deve risolvere un determinato problema. PROLOG è un linguaggio dichiarativo: tutte le sue istruzioni sono "dichiarazioni", enunciazioni di fatti. Quella che si costruisce, cosí, è fondamentalmente una grande base di dati, che si può poi andare a interrogare. Le istruzioni del PROLOG hanno una forma molto simile agli enunciati della logica matematica, in particolare a quelli del calcolo

La "tartaruga" (il cursore a forma di triangolo) ha disegnato un rettangolo e, a destra, una figura piú complessa. Questo è il tipico ambiente grafico del LOGO, linguaggio di programmazione potente ma semplice, derivato dal LISP (linguaggio diffuso negli studi sull'intelligenza artificiale) e particolarmente adatto per l'uso da parte di un bambino e in ambiente scolastico.

dei predicati: chi ha una qualche preparazione in questo settore può trovarsi perfettamente a proprio agio.

È stata realizzata anche una versione di PROLOG adatta per i personal computer, il MICROPROLOG. Non ha la potenza del fratello maggiore, ma è uno strumento davvero interessante.

Per finire questa veloce carrellata (che non ha certo pretese di esaustività), il LOGO: un linguaggio derivato dal LISP, ma rivolto in modo particolare al mondo dei bambini e dei ragazzi. Il suo ideatore, Seymour Papert, oggi a capo del Laboratorio di Intelligenza Artificiale del Massachusetts Institute of Technology, ha lavorato a lungo con Jean Piaget, e ha abbracciato molte idee fondamentali della sua pedagogia e della sua psicologia, in particolare quelle del gioco come strumento di apprendimento e dell'importanza dell'esplorazione individuale come mezzo conoscitivo. LOGO è un linguaggio che consente di lavorare con la matematica dal di dentro: il cursore è sostituito da un simbolo che sta per una tartaruga (originariamente si trattava veramente di un piccolo robot, non di un artificio sullo schermo del calcolatore), alla quale si danno comandi come AVANTI, INDIETRO, SINISTRA, DESTRA, con indicazioni del numero dei passi che deve compiere o dell'ampiezza dell'angolo di cui deve ruotare. Tutte le figure geometriche sono viste dall'interno, anziché dall'esterno come nella geometria tradizionale: un quadrato non è piú una figura con quattro lati e quattro angoli, ma il risultato di una procedura ripetitiva che sposta la tartaruga in avanti di un certo numero x di passi, la fa ruotare di 90 gradi, la porta avanti dello stesso numero x di passi, la fa nuovamente ruotare di 90 gradi, e via di seguito. L'esempio è elementare, ma la potenza del LOGO non si ferma certo qui: cambia il modo di fare goemetria, ma è possibile avanzare su questo terreno fino alla teoria della relatività. Semplicemente non si vedono piú, come nel modo tradizionale, le proprietà globali, ma si lavora sempre con proprietà locali.

Il LOGO (e qui si vedono le sue origini nel mondo del LISP) ha anche buone capacità di elaborazione di stringhe di caratteri e di liste, ma quello che risulta piú affascinante sono le sue capacità grafiche, invocate da meccanismi di programmazione semplici e intuitivi. È un linguaggio splendido da usare nella scuola, perché rende facile "fare cose" facendo al contempo dell'informatica.

Il LOGO è un linguaggio per i calcolatori personali: la cosa interessante è che ne sono state realizzate anche versioni in italiano, per le principali macchine in commercio. A differenza di tutti gli altri linguaggi di programmazione, cioè, le parole chiave (le "primitive" del linguaggio) in questi casi sono parole italiane: il che ne rende l'ingresso in classe molto piú semplice.

Oggi e domani

Sono bastati pochi anni perché, dai primi personal computer degli hobbisti (e per hobbisti) dell'elettronica, con parole di quattro bit, 4 K byte di memoria interna e un registratore a cassette come unità di memoria esterna, si passasse a macchine raffinate, con microprocessori a 8, poi a 16, ormai in qualche caso addirittura 32 bit, memorie interne che possono arrivare ai 1000 K byte o anche oltre, memorie esterne a dischetti flessibili, poi a dischi rigidi, in grado di archiviare decine di milioni di caratteri. E, parallelamente, la qualità del software si è elevata, al punto che ormai molti programmi non hanno nulla da invidiare qualitativamente ai programmi sviluppati per i grandi calcolatori *mainframe*. Il Visicalc della Visicorp ha addirittura introdotto uno strumento di elaborazione personale fino a quel punto sconosciuto nel mondo della grande informatica: se i primi personal computer erano poco piú di giocattoli divertenti, con un patrimonio di programmi elementari e poco ambiziosi, oggi esistono macchine potenti per le quali è disponibile software costruito professionalmente, in grado di sfruttare a pieno quella potenza, e pensato espressamente per l'uso personale.
Che cosa succederà ora? Non c'è bisogno di essere indovini per prevedere che, almeno ancora per un po' di tempo, continuerà a manifestarsi la medesima tendenza: avremo macchine piú potenti a parità di prezzo, se non a prezzi inferiori, e software sempre piú perfezionato, ma con tutta probabilità senza le stesse diminuzioni di prezzo (il costo del software va a incidere sempre di piú, in percentuale, sul costo totale di un sistema di calcolo personale: ma va anche detto che cresce parallelamente, almeno in media, anche la sua qualità). Ma fin qui, nulla di molto interessante. Anche la crescita della diffusione del calcolatore in ambito domestico non subirà bruschi salti, se nessuno avrà un'idea brillante, che crei anche per la casa un tipo di programma nuovo, soluzione efficiente per qualche problema ancora scoperto, cosí come i tabelloni elettronici hanno fatto da soli piú di qualsiasi altro programma, per la diffusione del personal computer in ambito professionale o nei piccoli (e grandi) uffici. Qualche evoluzione piú concreta, però, si può pronosticare senza troppi rischi.

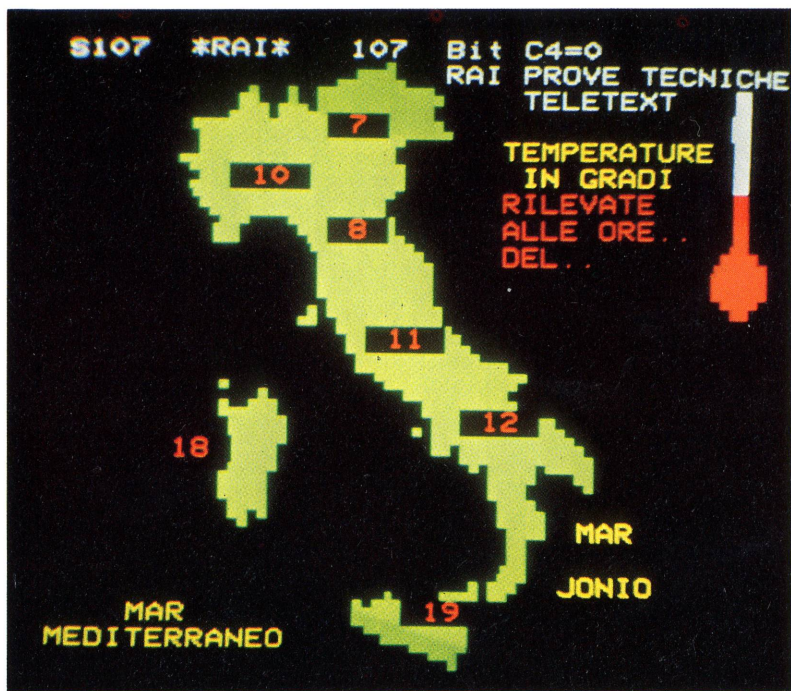
L'informatica è destinata a pervadere e a trasformare profondamente tutti i settori della vita sociale.

Le comunicazioni

L'uso prevalente del personal computer, oggi, è come strumento autonomo e indipendente: il che è anche giusto, vista la sua natura. Però è un peccato non sfruttare tutte le possibilità che apre l'abbinamento fra calcolatore e apparecchio telefonico! Il collegamento fra personal computer lungo le linee telefoniche permette lo scambio di materiale in modo rapido ed efficiente: dati, programmi, testi. Lo scambio di informazioni sotto forma di testi è possibile tra macchine di qualunque tipo: non ci sono limitazioni, purché si posseggano

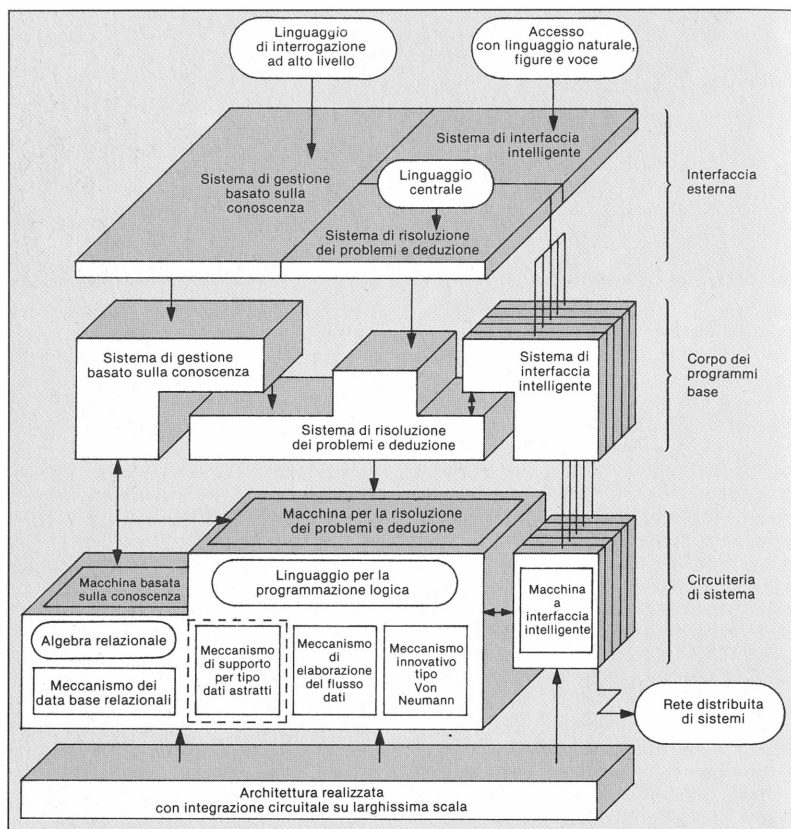
un modem o un accoppiatore acustico e un programma di comunicazione che permetta di trasferire le informazioni secondo un codice comune. Se le macchine in comunicazione sono diverse, non si può fare qualunque cosa: lo scambio di programmi sarà anche possibile, ma in genere inutile (ma se si usa un linguaggio di programmazione compilato, in forma standard, gran parte dei programmi sorgente potrebbero essere comunicati e utilizzati con successo).
Già questo permetterebbe di creare un piccolo sistema interpersonale di "posta elettronica": ma questo è un campo aperto ad altre possibilità, di altro respiro. Già esistono le prime forme di reti di vera e propria posta elettronica: una organizzazione – pubblica o privata – può creare una rete con un numero relativamente piccolo di grandi calcolatori dislocati nelle città principali, collegati fra loro da linee dedicate per la trasmissione di informazioni in forma digitale. Ciascuno, che possieda un telefono e un calcolatore, può affittare una "casella elettronica", che fisicamente non è altro che un pezzetto di memoria nel grande calcolatore che costituisce il "nodo" della città piú vicina. La casella è identificata univocamente da un "indirizzo": un numero, un nome, qualche sistema di identificazione. Chiunque abbia un computer e sia collegato alla rete può comunicare con il nodo della propria città e inviare alla mia casella qualunque materiale in forma digitale: deve solo formare il numero telefonico del nodo, chiedere il collegamento, precisare l'indirizzo di destinazione e inviare il messaggio. Si preoccupa poi il calcolatore del nodo a recapitare il messaggio a destinazione, eventualmente a centinaia di chilometri di distanza, nella mia casella elettronica. In qualunque momento io, a mia volta, posso collegarmi e chiedere di vedere i messaggi indirizzati a me che vi sono depositati. Il sistema è molto piú efficiente della posta tradizionale e dei corrieri trasportatori, e per grandi distanze molto piú veloce; fra l'altro può essere anche molto meno costoso, perché ogni utente della rete di posta elettronica deve effettuare collegamenti telefonici solo con il calcolatore del proprio nodo – nella propria città o nella città piú vicina che costituisca un nodo della rete – e quindi con costi di teleselezione nulli o abbastanza contenuti. Il costo di trasmissione da nodo a nodo è assorbito dai canoni di affitto della casella elettronica: non c'è vantaggio per qualche messaggio in arrivo o in partenza all'anno, ma per scambi piú frequenti il sistema è piuttosto interessante.
Delle banche dati abbiamo già avuto modo di dire qualcosa: anche questa è una possibilità sfruttata ancora poco, e interessante (almeno per ora) soprattutto per chi usa il personal computer a fini professionali; ma è prevedibile che anche nei paesi europei, come già è successo negli Stati Uniti, si vadano creando e ampliando progressivamente banche dati che archivino informazioni utili anche per le attività domestiche. Il servizio piú pregevole che si potrebbe immaginare sarebbe una banca di materiale didattico o, in generale, educativo, a cui poter attingere liberamente (a basso costo); ma il campo è aperto a tutte le idee e alla fantasia.

Il calcolatore in ufficio

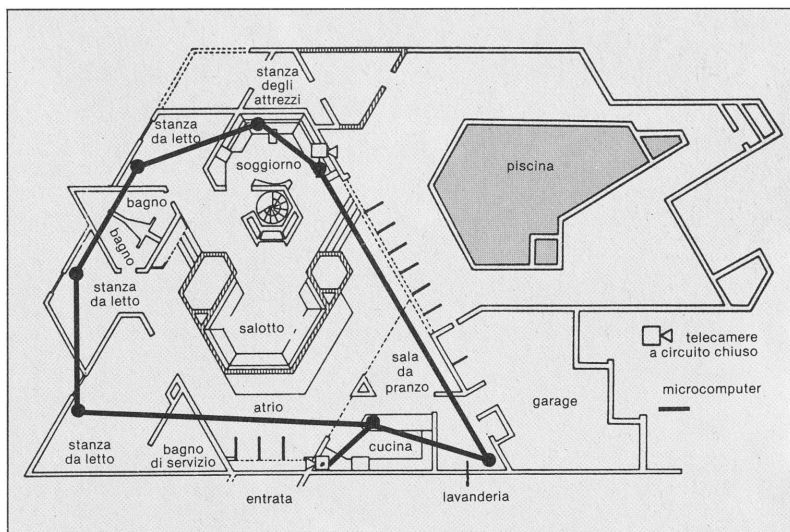
Il personal computer ha già fatto da tempo il suo ingresso negli uffici delle piccole e delle grandi aziende: nelle piccole scelto o proposto come soluzione di tutti i problemi gestionali, nelle grandi come strumento personale del manager che si rende indipendente dal centro di calcolo dell'azienda per le elaborazioni di supporto alla sua attività decisionale, ma al contempo – sempre attraverso il personal configurabile come terminale tradizionale – può ricevere dal centro di calcolo tutte le informazioni necessarie; o ancora come strumento del personale di segreteria, macchina per scrivere raffinata, eccellente per le circolari e per le relazioni importanti, dove anche un cambiamento all'ultimo minuto può essere incorporato facilmente, senza dover ribattere l'intero documento.
Molte piccole aziende sono rimaste pro-

Una "pagina" del servizio Televideo, gestito dalla RAI, nella fase di sperimentazione.

babilmente deluse dalla prima esperienza con il personal computer: il problema in genere non è del personal computer, ma dei rivenditori che non sanno valutare le esatte esigenze di un'azienda. In effetti il personal computer è nato come strumento di uso personale, non come strumento gestionale: per compiti di gestione sono necessarie caratteristiche marcate nell'elaborazione di archivi, aspetto sotto il quale i mini e i grandi elaboratori orientati agli usi amministrativi sono invece ben strutturati. È fondamentale non affidarsi a macchine troppo piccole: come minimo, anche per esigenze abbastanza limitate, è necessaria la possibilità di collegare un disco rigido. Anche una piccola azienda ha in fondo grandi quantità di informazioni da elaborare, e tutte le pratiche aziendali devono integrarsi: dall'archivio fornitori alla gestione di magazzino, all'archivio clienti, alla fatturazione, alle attività contabili generali. Sotto questo punto di vista, molto si risolverà con il crescere progressivo della potenza dei personal computer: già oggi, in fondo, i personal piú potenti sono in grado di far concorrenza ai minicomputer. Piú significativo, comunque, dell'uso del personal computer a fini gestionali, e piú importante sotto il profilo delle possibili conseguenze sociali, sembra invece l'uso come stazione di lavoro decentrata. È una tendenza già in atto negli Stati Uniti: meno impiegati in ufficio, piú persone che lavorano a casa propria, dotate di un proprio calcolatore personale in grado di fungere anche da terminale, in collegamento con il grande calcolatore dell'azienda. Il che può essere piacevole, per chi ama la propria casa e non ama gli orari d'uffi-

cio, ma può ovviamente anche favorire lo sviluppo di una nuova categoria di "cottimisti" – lavoratori non assunti ma pagati a quantità di lavoro svolto, con tutto quello che di precarietà può comportare. In una visione piú rosea delle cose, può significare per il singolo una maggiore possibilità di organizzare il proprio tempo e di lavorare secondo i propri ritmi anziché secondo ritmi impersonali e, per la comunità, una diminuzione del pendolarismo, un migliore sfruttamento degli ambienti costruiti, decongestionamento del traffico urbano, addirittura risparmio energetico. Una volta che i vari calcolatori siano collegati in rete, anche la presenza fisica diventa meno vincolante: si possono instaurare sedute periodiche di teleconferenza, in cui ciascuno può comunicare con gli altri mediante il proprio calcolatore e ricevere su di esso i vari messaggi; con il vantaggio di poter conservare sempre una registrazione di tutto quello che avviene. Il sistema di teleconferenza può essere attuato anche con mezzi televisivi abbinati alle linee telefoniche, anziché con calcolatori, oppure con una opportuna combinazione dei due tipi di mezzi: già oggi la tecnologia lo permette, anzi la teleconferenza con mezzi televisivi è già un servizio offerto (sia pure ancora in modo limitato) anche in Italia dalla SIP. Tutto que-

sto contribuisce a delineare una società in cui ci si sposta molto meno per motivi di lavoro, con minori perdite di tempo e minori consumi energetici: in fondo, perché correre in aereo per mille chilometri se lo stesso risultato può essere ottenuto stando tranquillamente a casa o in ufficio, usando il telefono, il computer per i messaggi scritti e un apparecchio televisivo per la visualizzazione dell'interlocutore?

I robot

Non hanno le forme umanoidi a cui ci hanno abituato i racconti e i film di fantascienza, ma sono già in mezzo a noi. Sono solo mani o bracci meccanici, ma sono veri automi programmabili. Molte industrie se ne sono dotate: le industrie automobilistiche, per esempio, li usano per la verniciatura o la saldatura a punti delle scocche d'automobile. In altre industrie sono utilizzati per adempiere a mansioni pericolose, o per operazioni di tipo manuale relativamente semplici e ripetitive.

Hanno cominciato a fare la loro comparsa anche i personal robot: come i personal computer si differenziano dai fratelli maggiori esclusivamente per le prestazioni, il prezzo e il fatto di essere rivolti all'uso personale, ma non per i principi di funzionamento. In effetti,

Il diagramma nella pagina a fronte mostra come sarà, secondo i progetti, la struttura di un calcolatore della quinta generazione (la cui costruzione effettiva è prevista per gli anni Novanta). L'interfaccia verso l'utente prevede modalità di comunicazione basate su segnali vocali e immagini grafiche; tutto il sistema dipende fortemente da tecniche nate nell'ambito delle ricerche sull'intelligenza artificiale. Nel disegno qui sopra, la pianta di una casa automatizzata (con una rete distribuita di microcalcolatori e largo uso di telecamere) progettata negli Stati Uniti da Charles Robert Schiffner.

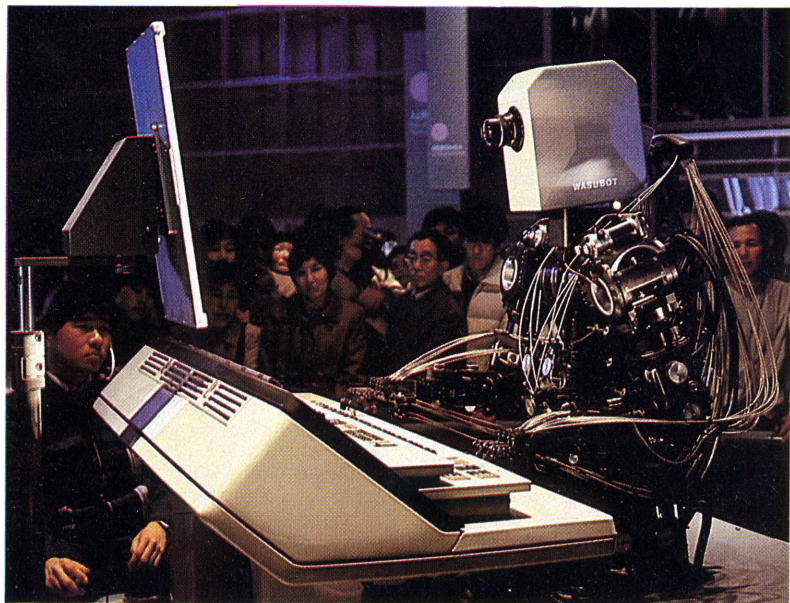
quelli che già esistono non sono ancora adatti a svolgere funzioni utili, in casa o in ufficio: sono soprattutto strumenti didattici, mezzi per familiarizzare con un nuovo settore tecnologico. Però alla fine degli anni Settanta più o meno la stessa cosa si diceva dei personal computer, quindi è lecito sospettare che l'evoluzione tecnologica ci porti presto dei personal robot funzionali.

I personal robot disponibili oggi sono sostanzialmente bracci meccanici, in

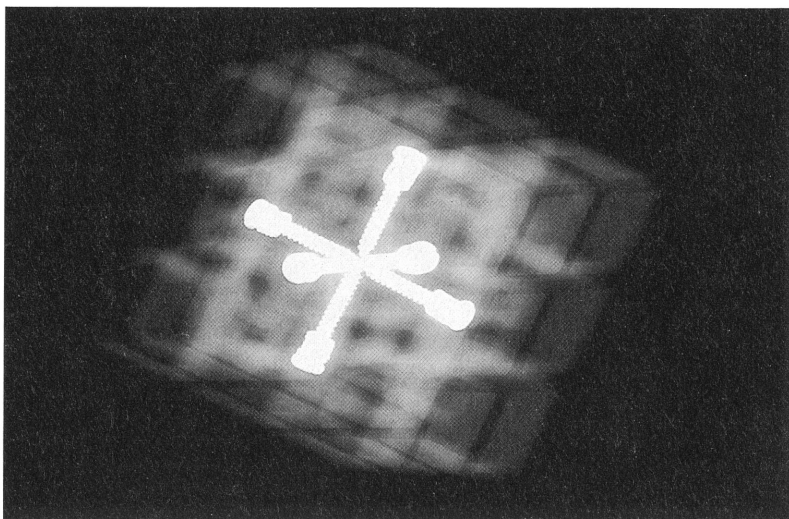
qualche caso mobili, in grado cioè di afferrare e sollevare un oggetto (il cui peso non deve superare i 500 grammi, mediamente) e di spostarlo, eventualmente spostandosi a propria volta mediante ruote o qualche altro meccanismo di locomozione. Capacità piú che sufficienti per un uso hobbistico e didattico.
Il personal robot ci interessa, in questo contesto, perché è un dispositivo programmabile: il suo "cuore" è un microprocessore che può essere programmato mediante un calcolatore personale, nei confronti del quale si comporta come un qualunque altro dispositivo esterno. È sufficiente, cioè, disporre di un'interfaccia adeguata (che in molti casi è un'interfaccia di comunicazione, del tipo RS-232) e del software di gestione, fornito in genere insieme al robot. Per chi vuole cimentarsi direttamente con la programmazione, esistono linguaggi specializzati; almeno uno dei robot disponibili commercialmente è programmabile in FORTH.
Qualche robot è già dotato di circuiti di sintesi vocale, grazie ai quali può pronunciare un piccolo numero di vocaboli (sempre, ovviamente, sotto controllo di programma): quello del riconoscimento del parlato e della sintesi vocale è, in effetti, uno dei campi ancora meno sviluppati e uno dei piú difficili da affrontare. La voce umana è uno strumento raffinatissimo, frutto di millenni di "ricerca e sviluppo" naturali attraverso i meccanismi dell'evoluzione, cosí come raffinatissimo è il nostro linguaggio. Solo per evidenziare un problema tra i molti aperti, bisogna tener presente che i segnali emessi da ciascuno di noi nel parlato non sono "uditi" solamente per il loro valore acustico: se si esamina il sonogramma di una "e" pronunciata da parlanti diversi, si può notare, al di là di qualche somiglianza generale, una for-

tissima variabilità. Una macchina deve basarsi fondamentalmente sulla struttura acustica del segnale vocale, ed è difficile "insegnarle" quali siano le differenze significative e quali invece quelle trascurabili. Per ora i metodi disponibili sono quasi "brutali", e richiedono un dispendio enorme di risorse di memoria: il riconoscimento della voce è realizzato in qualche prototipo di laboratorio, ma è ancora imperfetto e fortemente soggetto a errori (per essere utile deve essere "tarato" sulla voce dell'interlocutore, e anche allora presenta una notevole percentuale di incertezza). Il problema della sintesi è meno grave, se il vocabolario da sintetizzare è limitato: siamo invece ancora lontani dal disporre di sistemi che possano sintetizzare vocaboli a piacere, sulla base di una conoscenza della pronuncia dei singoli suoni. Cioè, tutto va bene finché si deve realizzare un sistema che dia l'ora esatta al telefono: le ore sono in tutto ventiquattro e i minuti sessanta, quindi il vocabolario è abbastanza ristretto. Ma anche i piú grandi calcolatori disponibili sono messi duramente alla prova nei tentativi di creare sistemi di sintesi vocale in grado di pronunciare un qualsiasi testo scritto introdotto, per esempio, tramite tastiera.

Di sintesi vocale può essere dotato qualunque calcolatore, non solo un robot; nel caso di un robot, semplicemente, sembra quasi una richiesta piú naturale. Ma altrettanto naturale è la richiesta di robot che presentino qualche capaci-

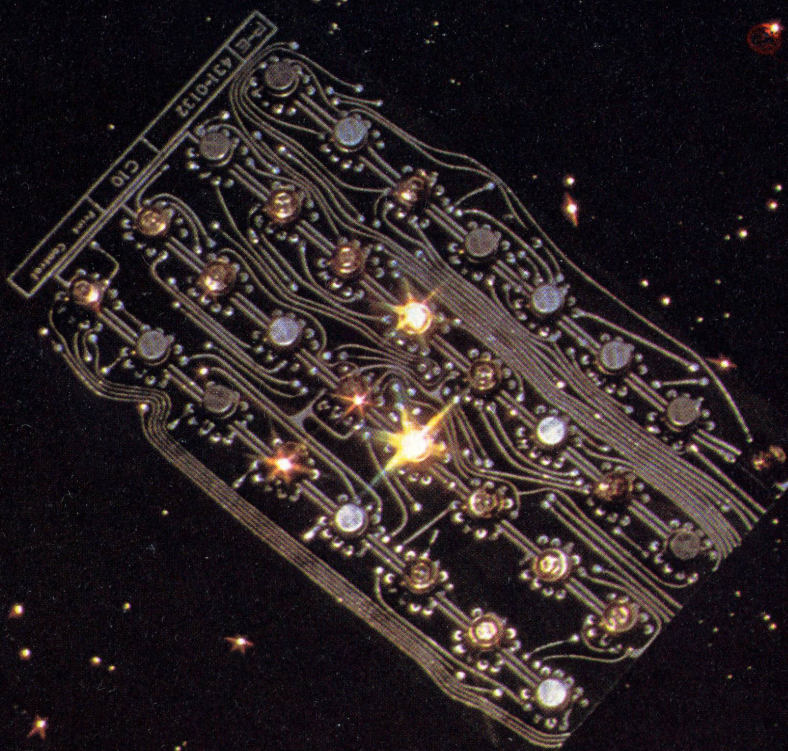
I personal robot non sono ancora macchine veramente utili, ma sono i primi prodotti di un mercato in rapida crescita, in grado di soddisfare esigenze hobbistiche o di piccola sperimentazione. Un piccolo robot personale come quello della pagina a fronte è in grado di muoversi su ruote o su qualche altro sistema di locomozione simile, ed è sostanzialmente un braccio meccanico in grado di afferrare e sollevare un oggetto non molto pesante (intorno ai 500 grammi). Viene programmato mediante un personal computer esterno, e il programma viene poi eseguito dal microprocessore interno al robot. Qui sotto, un robot pianista, in grado di leggere lo spartito e suonare, presentato all'Esposizione mondiale di Tsukuba, Giappone.

tà "visiva"; ma tanto è naturale per noi il processo della visione, che è stato quasi sconvolgente per i ricercatori scoprire quanta complessità sia nascosta in atti apparentemente semplici, che compiamo tutti i giorni con noncuranza, come riconoscere il volto di una persona in mezzo alla folla.
Insegnare a un calcolatore o a un robot "come si vede" è difficile: per registrare le immagini è sufficiente qualunque telecamera, ma il problema sorge dopo, quando si deve far esaminare alla macchina l'immagine, e si devono formulare algoritmi che consentano di identificare gli oggetti ripresi, di separarli gli uni dagli altri e dallo sfondo. Sono molti comunque i laboratori in tutto il mondo che si dedicano a ricerche in questi campi e i passi compiuti finora, anche se non hanno dato soluzioni definitive, hanno però permesso di realizzare macchine con qualche capacità di riconoscimento visivo, almeno in situazioni ben definite e artificiali.

L'intelligenza artificiale

Il riconoscimento della voce e la visione chiamano in causa prepotentemente processi di pensiero, processi che richiedono intelligenza: ma in fin dei conti, non molti anni addietro, chiunque avrebbe sostenuto che per effettuare dei calcoli complessi era necessaria l'intelligenza di un uomo, e ciononostante oggi possiamo avere sul nostro tavolo macchine che svolgono egregiamente quella funzione.
L'intelligenza dell'uomo è sempre presente: sono pur sempre uomini quelli che progettano queste macchine e che preparano i programmi che permettono loro di svolgere qualunque compito. È interessante tuttavia chiedersi fino a che punto si possa spingere questo processo. All'interno delle scienze del calcolatore, si è sviluppata, negli ultimi anni, una branca specifica, che studia i modi per costruire programmi che permettano a un calcolatore di svolgere compiti per i quali, se li svolge un uomo, diremmo che sia necessaria intelligenza. Gli aspetti piú affascinanti di questa disciplina sono quelli relativi ad attività non perfettamente regolate: attività nelle quali un uomo non applica semplicemente regole logiche precise ma si affida "all'intuito" o alla propria "esperienza" o al proprio "giudizio da esperto" o simili vaghe nozioni. Un buon giocatore di scacchi, per esempio, non si affida esclusivamente alla logica e a un'analisi brutale e completa di tutte le posizioni di gioco: è in grado, nella maggior parte delle situazioni, di decidere quasi a colpo d'occhio quali siano le mosse piú promettenti e si sofferma ad esaminare in dettaglio solo quelle. Costruisce tattiche e strategie, ed è capace di rinunciare a qualche pedina ora per raggiungere una posizione di maggior vantaggio piú avanti. Seguendo solo criteri piú rigidi, la macchina tende a condotte di gioco che possono apparire piú prudenti, ma alla lunga molto meno produttive.
Sono già stati realizzati programmi capaci di apprendere dall'esperienza, cioè di memorizzare gli esiti di situazioni precedenti e di sfruttare queste informazioni nella condotta futura: un obiettivo ambizioso è quello di formulare programmi in grado di modificare se stessi, sulla base dell'esperienza passata.
Esistono ormai sistemi a cui è stato dato il nome di "sistemi esperti" perché simulano il comportamento di un esperto in un particolare settore: i migliori sono nel campo della geologia e dell'esplorazione mineraria, altri sono nel campo della chimica o in quello della medicina. Un sistema esperto è costituito da una base di conoscenza e da regole deduttive ed euristiche. La base di conoscenza è una raccolta organizzata di informazioni, il piú possibile ampia, sull'argomento a cui il sistema è dedicato; le regole deduttive sono le regole logiche convenzionali e ben collaudate nel ragionamento scientifico; le regole euristiche sono invece regole di natura piú sfumata, che l'esperienza insegna possono costituire una buona guida in molti casi, anche se non in tutti e anche se non forniscono risposte sicure ma solo probabili, che tuttavia possono poi essere sottoposte a controlli con metodi piú rigorosi. Le regole euristiche sono

Il "cubo di Rubik" ai raggi X. È stato costruito anche un piccolo robot in grado di risolvere il cubo: impiega due decimi di secondo a trovare la soluzione, ma sei minuti per manipolare effettivamente il cubo.

quelle che ci permettono di comportarci con successo con gli altri, per esempio: non è assolutamente detto che se ci si comporta in un certo modo si può fare amicizia con una persona, ma nella maggior parte dei casi taluni atteggiamenti possono favorire il nascere di un rapporto amichevole. Si decide allora per quel comportamento: e qualche delusione ogni tanto non modificherà la nostra convinzione. Analogamente un buon medico è in grado di valutare i sintomi di un malato e di diagnosticare una malattia, anche se non ha in mano tutti gli elementi possibili e immaginabili per decidere, ma semplicemente perché può farsi guidare anche da conoscenze di tutt'altra natura (come il sapere, per esempio, che si sono verificati molti altri casi della stessa malattia nella zona, il che rende piú probabile una diagnosi in quel senso). Anche i medici esperti sbagliano, a volte, e i sistemi esperti basati sul calcolatore non pretendono di fare meglio: sono studiati per fornire ipotesi plausibili e per segnalare il grado di probabilità delle ipotesi stesse. Non devono soppiantare l'esperto, ma fornirgli un ausilio soddisfacente.

Qualche programma di sistema esperto esiste attualmente anche per i personal computer: si tratta di strumenti di uso personale, per lo piú con finalità didattiche, ma sono indicazioni precise della possibilità di sviluppo. Se non si potrà mai pretendere sul calcolatore di casa tutta la potenza dei sistemi studiati per i grandi calcolatori, si potrà sicuramente disporre comunque, in futuro, di programmi "intelligenti" creati su misura per l'uso personale.

So perfettamente che in questa veloce carrellata sono piú le cose tralasciate di quelle riportate: è solo uno spiraglio verso il futuro, un piccolo assaggio di quello che potrà diventare, e in parte già è, il mondo dell'informatica personale. Ma mi sembra che ci sia abbastanza di che mantener desto l'interesse per molti anni a venire. Sicuramente siamo ancora solo agli inizi: le sorprese e le avventure intellettuali che ci aspettano sono molte e difficilmente immaginabili.

Glossario

Accesso, metodo di (*Access method*) Metodo utilizzato dal sistema per recuperare un dato registrato in precedenza in una memoria magnetica, o viceversa per immagazzinarlo in una memoria magnetica. I piú comuni sono l'accesso diretto (→), l'accesso casuale (→) e l'accesso sequenziale (→).
Accesso, tempo di (*Access time*) Il tempo necessario per la lettura o la scrittura di informazioni in un sistema di memoria. Nelle memorie principali (interne alla macchina) i tempi di accesso sono molto brevi; sono molto piú lunghi per i sistemi di memoria secondaria (di massa), esterni alla macchina, come dischi o nastri magnetici. I sistemi ad accesso sequenziale hanno tempi di accesso mediamente piú lunghi di quelli ad accesso casuale o diretto.
Accesso casuale (*Random access*) Metodo di accesso alla memoria in cui il recupero e l'immagazzinamento delle informazioni sono fisicamente indipendenti dalla posizione delle informazioni precedenti e seguenti. È il metodo normalmente utilizzato per i dischi magnetici. Con l'accesso casuale, il tempo per l'accesso a qualunque informazione è indipendente dalla sua localizzazione.
Accesso diretto (*Direct access*) Metodo di accesso in cui recupero e immagazzinamento delle informazioni sono basati sull'indirizzo fisico della memoria, che coincide con una determinata posizione del braccio o della testina di lettura/scrittura.
Accesso sequenziale (*Serial access*) Metodo di accesso in cui recupero e immagazzinamento delle informazioni avvengono per blocchi successivi: in questo caso la posizione di una informazione è strettamente dipendente dalla posizione delle informazioni precedenti e successive. L'accesso sequenziale è quello tipico dei registratori a nastro: per recuperare un'informazione, bisogna scorrere tutto il nastro fino al punto in cui si trova quell'informazione. L'accesso casuale ha mediamente tempi molto lunghi, ma variabili in funzione della posizione dell'informazione. Si può pensare alle normali cassette audio: per trovare l'ultima canzone bisogna far scorrere tutto il nastro fino verso la fine; su un disco, invece, dove l'accesso è di tipo casuale, ci vuole sempre lo stesso tempo a posizionare il braccio su qualunque brano.
Accoppiatore acustico (*Acoustic coupler*) Un dispositivo, analogo al modem, che converte dati digitali in segnali acustici (due suoni di frequenza diversa, corrispondenti a 0 e 1), che possono poi essere trasmessi lungo una linea telefonica, come una comunicazione vocale e, viceversa, riceve segnali acustici dalla linea telefonica e li ritrasforma in segnali digitali per un computer. La trasmissione mediante accoppiatore acustico è meno affidabile di quella mediante modem, tuttavia questo dispositivo ha il vantaggio di poter essere trasportabile e quindi di permettere la trasmissione di dati da qualunque apparecchio telefonico.
Accumulatore (*Accumulator*) Un registro nell'unità aritmetica e logica (ALU), usato soprattutto per le operazioni aritmetiche e per istruzioni attinenti ad attività di input/output.
ADA Un linguaggio di programmazione di alto livello, sviluppato alla fine degli anni Settanta sotto l'egida del Dipartimento della Difesa degli Stati Uniti: il nome è un omaggio ad Ada Lovelace, figlia di lord Byron e assistente di Charles Babbage, che può essere considerata la prima "programmatrice" della storia. Il linguaggio ADA è molto potente, ed è pensato come un linguaggio di uso il piú generale possibile. Per il momento, è un linguaggio confinato ai grandi sistemi.
ADC → Convertitore A/D.
AI → Intelligenza artificiale.
ALGOL Acronimo per ALGOrithmic Language, linguaggio algoritmico. Un linguaggio di programmazione di alto livello, progettato in particolare per applicazioni matematiche. È stato il principale linguaggio accademico, per l'insegnamento della programmazione, degli anni Sessanta. I programmi in ALGOL sono organizzati a blocchi e possono anche essere ricorsivi.
Algoritmo (*Algorithm*) Una procedura completamente specificata, passo per passo, per la risoluzione di un problema e, piú in generale, per l'esecuzione di una attività. La parte essenziale di un programma per calcolatore è un algoritmo, espresso in un linguaggio appropriato, che "dice" alla macchina, passo per passo, tutto quello che deve fare per produrre il risultato desiderato.
Alternativa → Struttura di controllo.
ALU (*Arithmetic Logic Unit*) Sigla che indi-

ca l'unità logica e aritmetica, una delle parti costitutive della CPU (→). È l'unità in cui vengono effettivamente svolte le informazioni di carattere logico e aritmetico, cioè le operazioni booleane e quelle di somma, sottrazione, moltiplicazione, divisione, ecc.

Analogico (*Analogic*) Si dice di ogni sistema che rappresenti le informazioni mediante grandezze che possono variare con continuità: è opposto a digitale (→). Un orologio tradizionale ha lancette che segnano l'ora con continuità; un orologio digitale in forma numerica, per unità discrete.

AND Uno dei circuiti logici fondamentali di un sistema di elaborazione e uno degli operatori booleani fondamentali. AND in inglese significa "e", e il comportamento del circuito e dell'operatore è analogo a quello della congiunzione: dà risultato vero (1) se e solo se ambedue i segnali in ingresso (ambedue gli operandi) sono 1; dà risultato falso (0) in ogni altro caso.

Architettura (*Architecture*) La costruzione di un sistema di elaborazione, vista a un livello generale: l'insieme delle sue istruzioni fondamentali, l'interfaccia verso l'utente, l'organizzazione della memoria, le operazioni fondamentali per l'ingresso e l'uscita e simili.

ASCII Sigla di American Standard Code for Information Interchange (codice americano standard per lo scambio di informazioni). Il piú diffuso fra i codici internazionali che rappresentano numeri, lettere e simboli per mezzo di valori binari univoci, comprensibili ed elaborabili dal computer. È anche il principale fra i codici di trasmissione dati fra sistemi di elaborazione. I personal computer utilizzano normalmente il codice ASCII (oppure un suo sottoinsieme).

Assemblatore (*Assembler*) Un programma "traduttore" che traduce in linguaggio macchina le istruzioni scritte in un linguaggio di programmazione di basso livello (detto linguaggio di assemblatore o linguaggio assembly), trasformando i codici simbolici delle operazioni nelle relative istruzioni proprie del computer, assegnando indirizzi di memoria alle istruzioni.

Assemblatore, linguaggio di (*Assembly language*) Un linguaggio di programmazione simbolico di "basso livello", ancora molto vicino al linguaggio macchina, legato cioè alle caratteristiche fisiche del computer su cui deve essere utilizzato e, di conseguenza, non trasportabile da una macchina all'altra. Viene tradotto in linguaggio macchina da un assemblatore, e fra le istruzioni simboliche e le istruzioni in linguaggio macchina esiste un rapporto "uno a uno" (un'istruzione simbolica corrisponde a una istruzione di macchina).

Assembly, linguaggio → Assemblatore, linguaggio di.

Base di dati (*Data base*) Una raccolta generalizzata e integrata di dati relativi a un'azienda, a un campo del sapere, a una organizzazione ecc., destinata a soddisfare le esigenze di tutte le applicazioni che vi hanno accesso e organizzata in modo da riprodurre le relazioni "naturali" che intercorrono fra i dati in essa archiviati. Una base di dati in generale serve a piú applicazioni e ha esistenza e validità indipendenti dai singoli programmi applicativi. Con il diffondersi dei piccoli computer, il termine (e il suo equivalente inglese) hanno assunto il significato piú generico di qualunque raccolta di dati relativamente omogenei fra loro, memorizzati su supporto magnetico in modo da renderne agevole il recupero selettivo in funzione delle esigenze dell'utente.

Base di dati distribuita (*Distributed data base*) In un sistema informativo distribuito, base di dati unica da un punto di vista logico, ma suddivisa fisicamente fra piú sedi.

BASIC Sigla di Beginner's All-purpose Symbolic Instruction Code (codice di istruzioni simboliche di uso generale per principianti), indica un linguaggio di programmazione di alto livello. Il BASIC è il linguaggio piú diffuso sui personal computer; per la sua relativa semplicità può essere appreso rapidamente, e, per il fatto di essere un linguaggio interpretato, permette un'interazione diretta fra l'utente e la macchina che, soprattutto per i principianti, risulta molto efficace. Come dice il nome, è un linguaggio di uso generale: il che significa che non è stato progettato per compiti specifici, e può essere utilizzato con pari risultati in tutti i campi (ma allo stesso tempo risulta in tutti un po' limitato). Esistono del BASIC numerosi "dialetti", che coincidono negli elementi fondamentali, ma differiscono profondamente per altri, in particolare per caratteristiche dipendenti dalla macchina come la grafica, il colore, la gestione dei suoni.

Baud Unità di velocità di trasferimento di dati. Normalmente si definisce come una velocità di trasferimento di un bit al secondo. "Trasmettere a 2400 baud", per esempio, significa trasmettere informazioni alla velocità di 2400 bit al secondo.

BCD Sigla di Binary Coded Decimal (decimale codificato in binario), indica un metodo di codificazione dei numeri decimali, utilizzato per la memorizzazione in un computer. Anziché i numeri, vengono codificate le cifre: una sequenza di 0 e 1 corrisponde all'1 decimale, un'altra al 2, una al 3, e via dicendo. È una codificazione meno efficiente rispetto al sistema di numerazione binario puro, e in particolare richiede un maggiore spazio di memoria e rende piú lenta l'esecuzione delle operazioni aritmetiche, tuttavia permette di ottenere risultati di grande precisione.

Benchmark Letteralmente, punto di riferi-

mento. Un test o una serie di test per valutare le prestazioni di prodotti hardware o software in confronto con altri o isolatamente. Non esistono tecniche di benchmark soddisfacenti per tutte le categorie di prodotti e per tutti gli aspetti dei prodotti: si tratta di un campo che attende ancora una valida sistemazione metodologica.

Bidirezionale (*Bidirectional*) Che procede in due direzioni. Si dice tipicamente delle stampanti che stampano i caratteri sia quando il meccanismo di stampa procede da sinistra verso destra, sia quando procede da destra verso sinistra.

Binario, sistema (*Binary system*) Sistema di numerazione posizionale a base 2: usa esclusivamente le cifre 0 e 1 e ogni posizione, in un numero di piú cifre, indica una potenza 2, cosí come nel sistema decimale ogni posizione indica una potenza di 10.

Bistabile flip-flop ➔ Flip-flop.

Bit di parità (*Parity bit*) Un bit in piú che viene aggiunto a un byte, al fine di poter controllare se i contenuti del byte sono stati trasmessi o immagazzinati correttamente. Ove la parità sia di tipo dispari, il bit di parità viene fissato a 1 o a 0, secondo i casi, in modo che la somma dei bit che costituiscono il byte risulti un numero dispari; in caso la parità scelta sia invece di tipo pari, si fissa un bit di parità a 1 o a 0 in modo che la somma dei bit che costituiscono il byte sia un numero pari. Se, in un controllo successivo, i bit di un byte, sommati al relativo bit di parità, non danno somma pari (o dispari) secondo quanto determinato dal tipo di parità scelto, si può sapere che è sopravvenuto un errore.

Bit di stop (*Stop bit*) Nella trasmissione di dati, bit usato per la sincronizzazione, che segue ogni byte trasmesso in modo asincrono attraverso una porta seriale. Alcuni sistemi richiedono anche piú di un bit di stop, ma ambedue gli estremi di ciascun collegamento di comunicazione debbono utilizzare lo stesso numero di bit di stop, che vengono automaticamente aggiunti alla sorgente ed eliminati alla destinazione per mezzo di appositi circuiti hardware.

Booleana, algebra (*Boolean algebra*) Un particolare tipo di algebra che viene utilizzato nella definizione del comportamento dei circuiti fondamentali di un calcolatore. Un'algebra booleana è un insieme di elementi su cui sono definite tre operazioni: le operazioni binarie AND e OR (chiamate anche prodotto e somma) e l'operazione unaria NOT (o complemento). Quella usata nel campo dei calcolatori è un'algebra booleana particolare, con due soli elementi: 0 e 1 (complemento uno dell'altro), che coincide con l'algebra delle proposizioni (o calcolo proposizionale), quel ramo della logica che studia formalmente il valore di verità delle proposizioni composte in funzione dei valori di verità delle proposizioni elementari.

Booleana, funzione (*Boolean function*) È una funzione nell'algebra booleana (viene chiamata anche funzione logica): una combinazione di valori di verità mediante gli operatori dell'algebra.

Bootstrap L'operazione di "lancio" automatico, all'accensione di un computer, di un programma minimale, memorizzato in ROM, che attiva le funzionalità di base della macchina e, in genere, determina il caricamento di un programma piú complesso, il sistema operativo, che gestisce le risorse della macchina e la mette in condizioni di poter accettare ingressi dall'esterno.

Buffer o **Memoria di transito** (*Buffer*) Memoria ausiliaria (detta a volte anche memoria tampone), per memorizzare momentaneamente dati o per equilibrare l'attività di unità collegate che funzionano a velocità diverse. L'unità centrale di elaborazione, per esempio, è molto piú veloce di qualsiasi stampante: quando invia in uscita dati per la stampa, li invia alla sua velocità caratteristica a un buffer che poi li scarica verso la stampante a una velocità adatta a questa periferica, mentre l'unità di elaborazione può tornare a svolgere altre operazioni.

Bug Nel gergo degli informatici, un errore, in genere in riferimento a un programma; nella lingua inglese il termine (che letteralmente significa "insetto") è usato a volte anche per i guasti dell'hardware.

Bus Pista, canale su cui si trovano a viaggiare i segnali da una unità all'altra di un sistema di elaborazione.

C Un linguaggio di programmazione di alto livello sviluppato ai Bell Laboratories da Dennis Ritchie come linguaggio per la programmazione di sistema per il sistema operativo UNIX, sul minicalcolatore PDP 11/70 della Digital Equipment Corporation. Gli obiettivi nella costruzione del linguaggio erano l'ottimizzazione del tempo di esecuzione, delle dimensioni e dell'efficienza dei programmi. Eccellente nella programmazione di sistema, si è dimostrato valido anche come linguaggio di uso generale, ed ora è disponibile anche su calcolatori personali. È relativamente piccolo, compatto e di facile apprendimento.

CAD Sigla di Computer-Aided Design, progettazione assistita da calcolatore. Indica l'uso del computer nelle fasi di progettazione di un prodotto, con l'utilizzazione di una base di conoscenze, memorizzate elettronicamente, relative ai processi industriali interessati. La fase di progettazione può essere anche integrata alla fase di produzione controllata da calcolatore: i dati di progettazione possono

essere trasferiti a macchine a controllo numerico, in modo da guidare opportunamente anche i processi produttivi.

CAI Sigla di Computer-Aided Instruction, istruzione assistita da calcolatore. Indica l'uso del computer in generale nelle attività educative, in particolare in ambito scolastico.

CAM Sigla di Computer-Aided Manufacturing, produzione assistita da calcolatore. Indica l'insieme delle tecniche, basate sul calcolatore, per la gestione del processo produttivo in forma integrata: comprende quindi sia il controllo di processo, sia altri aspetti, come per esempio l'allocazione delle risorse, la gestione delle scorte, l'ordinazione dei materiali e via dicendo.

Carattere (*Character*) Qualunque tipo di simbolo (cifre, lettere dell'alfabeto, segni di interpunzione, spazio vuoto, simboli matematici e simboli grafici speciali) accettato in ingresso dal calcolatore.

Cartuccia (*Cartridge*) Un astuccio di plastica contenente una ROM, in cui sono registrati programmi, dotata di un connettore a pettine per il collegamento a un calcolatore. Costituisce un sistema sicuro e relativamente poco costoso per la commercializzazione di software, utilizzato soprattutto nel campo dei piccoli elaboratori domestici.

Catasta (*Stack*) Una porzione di memoria organizzata come una catasta di vassoi in un self-service: tutto avviene sempre sulla cima. A ogni istante, la prima informazione che può essere recuperata è l'ultima memorizzata: è un meccanismo che si definisce "ultimo dentro, primo fuori". La catasta (chiamata anche "pila") può essere implementata direttamente via hardware, oppure può essere solo una configurazione determinata dal software.

Chip Una piastrina di silicio su cui sono alloggiati i circuiti integrati a larga scala di integrazione. Su un'area di pochi millimetri quadrati possono essere contenuti migliaia di componenti elettronici.

Clock Letteralmente, "orologio". È reso meglio, in italiano, con "sincronizzatore" o "temporizzatore". Un circuito elettronico che fornisce i segnali di sincronizzazione per tutte le attività interne a un computer. Un ciclo di clock è il tempo minimo impiegato dall'hardware della macchina per l'esecuzione di una operazione elementare. La velocità del sincronizzatore è data normalmente in Hertz, per lo piú in megahertz (MHz, milioni di hertz). 1 hertz è pari a un ciclo al secondo. Una macchina che disponga di un clock a 4 MHz è in grado di eseguire 4 milioni di operazioni elementari al secondo.

COBOL Acronimo di COmmon Business Oriented Language (linguaggio comune orientato agli affari), indica un linguaggio di programmazione di alto livello, orientato specificamente alla programmazione per le attività commerciali, sviluppato nel corso degli anni Sessanta.

Codice (*Code*) Una regola o un insieme di regole per convertire un messaggio da una forma di espressione a un'altra. È un codice quello del telegrafo, che trasforma le lettere dell'alfabeto in linee e punti; è un codice quello delle bandiere di segnalazione sulle navi. Codice binario, in particolare, è qualunque codice che converta il messaggio in un alfabeto formato da due soli simboli, ancor piú in particolare in 0 e 1 o in segnali presenza/assenza di tensione.

Comando (*Command*) Una istruzione di un linguaggio di programmazione interpretato usata in modo immediato, cioè per "dire" al computer di eseguire immediatamente un'operazione. La maggior parte delle istruzioni del BASIC possono essere usate come comandi, in modo immediato: alcune hanno senso invece solo in modo differito, usate cioè all'interno di un programma. Quando un'istruzione è usata in modo differito non è piú indicata come "comando" ma come "enunciato" o, all'inglese, "statement".

Compatibilità (*Compatibility*) Si dice che due computer sono compatibili se i programmi scritti per l'uno possono essere eseguiti anche sull'altro. La compatibilità totale fra due sistemi è rara: piú spesso si limita ad alcuni aspetti, e non è detto che sia biunivoca, che cioè anche i programmi scritti sul secondo possano essere eseguiti sul primo.

Compilatore (*Compiler*) Un programma "traduttore" per linguaggi di alto livello: traduce programmi scritti in linguaggi di alto livello (programmi sorgente) in programmi scritti in codice di macchina (programmi oggetto). La traduzione viene effettuata una sola volta (fase di compilazione) su tutto il programma: quella che viene eseguita è poi sempre la versione compilata. I linguaggi per cui esiste un compilatore si dicono linguaggi compilati (in contrapposizione ai linguaggi interpretati, tradotti da un interprete). A differenza di un interprete, un compilatore può rendere faticosa la costruzione di un programma: se in fase di compilazione viene identificato un errore, il programmatore deve riprendere in considerazione (usando un apposito programma per la stesura di programmi sorgente, un text editor tipicamente) il programma sorgente nel suo complesso, poi tornare a compilarlo interamente. In compenso, l'esecuzione dei programmi, una volta compilati, è molto piú veloce rispetto all'esecuzione dei programmi scritti in linguaggi interpretati.

Computer graphics Denominazione anglosassone, entrata nell'uso generale, per indicare tutte le tecniche di grafica al calcolatore, per la produzione di disegni e immagini,

emessi in uscita su video o su stampante, in forma monocromatica o a colori.

Concentratore (*Concentrator*) Apparecchiatura che consente di far confluire i dati provenienti da un certo numero di collegamenti o linee di trasmissione in un minor numero di collegamenti a maggiore capacità, e di smistare il flusso dalle linee o collegamenti a grande capacità sui collegamenti minori.

Conversione (*Conversion*) Qualunque procedimento per il passaggio da una forma di rappresentazione a un'altra. Conversione, per esempio, è il passaggio dal sistema di numerazione decimale a quello binario o viceversa, o la trasformazione di un valore dal sistema metrico anglosassone al sistema metrico decimale.

Convertitore A/D (*A/D Converter*) Un dispositivo in grado di trasformare segnali analogici (A), cioè di tipo continuo, in segnali digitali (D), cioè di tipo numerico, discreto. Costituisce una forma di dispositivo di ingresso per un calcolatore, ed è usato tipicamente in applicazioni come il controllo di processo, dove è necessario disporre di dati relativi a una grandezza che varia con continuità, come la pressione o la temperatura, o in applicazioni grafiche e musicali.

Convertitore D/A (*D/A Converter*) Un dispositivo in grado di trasformare segnali digitali in segnali analogici: costituisce un tipo di dispositivo di uscita per un computer, usato tipicamente in tutti i casi in cui il computer deve essere utilizzato per pilotare apparecchiature che accettano in ingresso solo segnali continui. È il caso, per esempio, di robot o attuatori in applicazioni di controllo di processo, oppure degli amplificatori audio nel caso della sintesi musicale da calcolatore.

CPI Characters Per Inch, caratteri per pollice. Una misura della densità di stampa di una stampante.

CPS Characters Per Second, caratteri al secondo. Una misura della velocità di stampa di una stampante.

CPU Sigla di Central Processing Unit, unità centrale di elaborazione. È la parte di un calcolatore in cui vengono eseguiti effettivamente tutti i processi di elaborazione, e consta fondamentalmente di una unità aritmetica e logica (ALU) per l'esecuzione delle operazioni aritmetiche e dei confronti logici, di una unità di memoria veloce e di una unità di controllo che gestisce tutto il flusso di informazioni fra le unità di ingresso e di uscita, l'unità aritmetica e logica e l'unità di memoria interna.

CRI Cathode Ray Tube, in italiano tubo a raggi catodici.

Cursore (*Cursor*) Un rettangolino o una lineetta luminosa, fissi o lampeggianti, che compaiono sullo schermo di una unità video in funzione e indicano il punto in cui verrà visualizzato il successivo intervento dell'operatore; per esempio, dove verrà visualizzato il successivo carattere battuto alla tastiera, o in cui verrà effettuata la successiva correzione, ecc.

DAC → Convertitore D/A.

DBMS Sigla per Database Management System, sistema per la gestione di base di dati. Indica un software integrato che definisce la struttura fisica e logica dei dati in una base di dati, e fornisce gli strumenti per accedere ai dati stessi, inserirli, aggiornali, recuperarli.

Debugging Nel gergo, togliere i "bugs", gli errori da un programma. È una delle fasi piú importanti e faticose nella realizzazione del software. L'espressione italiana piú usata con lo stesso significato è "messa a punto".

Dedicato (*Dedicated*) Si dice di un calcolatore e, piú in generale, di una apparecchiatura elettronica digitale progettata per svolgere un compito ben determinato in modo esclusivo o prevalente. Esistono calcolatori specializzati per giochi, per l'elaborazione di testi (word processor), ecc. Una macchina dedicata "sa fare" una sola o poche cose, ma quel che sa fare lo fa molto bene, in genere con maggiore velocità o maggiore efficienza di una macchina di uso generale in cui si possa eseguire un programma equivalente.

Diagramma di flusso (*Flow chart*) Un particolare tipo di diagramma usato per visualizzare una procedura, un algoritmo, i cui elementi fondamentali sono caselle di varia forma, collegate da frecce. La forma della casella simbolizza il tipo di operazione che si svolge a quel passo: un'operazione di input, una decisione, e simili. Le frecce indicano i collegamenti tra i passi della procedura. I diagrammi di flusso sono utilizzati nelle prime fasi della programmazione, per evidenziare i singoli passi del procedimento di soluzione di un problema, prima di codificarli in un particolare linguaggio di programmazione.

Digitale (*Digital*) Si dice di ogni sistema basato su una rappresentazione dell'informazione mediante elementi discreti (numeri, simboli, ecc.). È opposto ad analogico (→).

Disco rigido (*Hard disk*) Un tipo di memoria di massa, costituito da un disco di alluminio rivestito di materiale magnetico e sigillato in un apposito contenitore insieme con la meccanica di lettura e registrazione. Un disco rigido può conservare una quantità di informazioni di gran lunga maggiore rispetto a un disco flessibile e presenta tempi di accesso inferiori.

Drive Unità di lettura e scrittura (registrazione) di informazioni su disco magnetico.

EAROM Sigla di Electrically Alterable Read-Only Memory, memoria a sola lettura

elettricamente modificabile. Un tipo di memoria a sola lettura (ROM) che può essere cancellata e riprogrammata piú volte. È simile alla EPROM, ma viene cancellata mediante corrente elettrica, anziché mediante luce ultravioletta. È una memoria costosa, usata in applicazioni particolari, come il controllo di macchine utensili, dove è periodicamente necessario modificare completamente i programmi di base.

EDP Electronic Data Processing, elaborazione di dati mediante apparecchiature elettroniche.

EFT Electronic Funds Transfer. → Trasferimento elettronico di fondi.

Elaborazione di testi (*Word processing*) Denominazione generale per tutte le attività, condotte con una apparecchiatura elettronica automatica, relative alla preparazione, alla correzione, alla stampa di testi e documenti di qualunque genere. Può avvenire mediante macchine specificamente dedicate, o con calcolatori di uso generale, grazie ad appositi programmi.

ENIAC Electric Numeral Integrator and Calculator, calcolatore e integratore numerico elettrico. Primo calcolatore elettronico digitale di uso generale, costruito all'Università della Pennsylvania, entrato in funzione nel 1945.

EOM End of Message, fine del messaggio. Nelle comunicazioni, codice di controllo internazionale che indica la conclusione della trasmissione di uno o piú testi, che formano un messaggio completo.

EOT End of Transmission, fine della trasmissione. Nelle comunicazioni, codice di controllo internazionale che indica la conclusione di una sequenza di trasmissione e riporta in condizione di riposo i terminali o le apparecchiature interessate.

EPROM Erasable Programmable Read Only Memory, memoria a sola lettura programmabile e cancellabile. È un tipo di ROM i cui contenuti possono essere cancellati e riprogrammati, per mezzo di opportune apparecchiature. Per la cancellazione si usa luce ultravioletta.

Esadecimale, sistema (*Hexadecimal system*) Sistema di numerazione posizionale a base 16, che usa come cifre le dieci cifre del sistema decimale e le prime sei lettere dell'alfabeto (A corrisponde a 10, B a 11, C a 12, D a 13, E a 14 e F a 15). Come nel sistema decimale ogni posizione rappresenta una potenza di 10, allo stesso modo nel sistema esadecimale ogni posizione rappresenta una potenza di 16.

EXT End of Text, fine del testo. Nelle comunicazioni, codice di controllo di trasmissione che viene utilizzato per indicare la conclusione di un testo.

File Letteralmente, cartelletta da mettere in archivio. Ogni blocco di informazioni che viene memorizzato e recuperato come singola unità, nella memoria di un computer.

Filtro (*Filter*) Dispositivo per il controllo delle frequenze che possono passare in un circuito: consente alle frequenze selezionate di transitare senza attenuazione significativa, e blocca le frequenze indesiderate. In un filtro passa alto è consentito il passaggio a tutte le frequenze al di sopra di un valore di soglia (frequenza di taglio); in un filtro passa basso è consentito il passaggio a tutte le frequenze al di sotto di una determinata frequenza di taglio. Nel filtro passa banda è consentito il passaggio alle frequenze comprese entro una coppia di valori, mentre nel filtro a reiezione di banda vengono lasciate passare tutte le frequenze esterne a una coppia di valori.

Flip-flop Un elemento circuitale elettronico, che è in grado di assumere due soli stati, di commutare dall'uno all'altro secondo modalità ben precise e riproducibili e, in mancanza di impulsi di commutazione, rimane stabilmente nello stato in cui si trova. Il tipo piú semplice di flip-flop può essere realizzato con due sole porte logiche di tipo NAND: il comportamento del circuito (indicato come flip-flop RS) dipende sia dall'ingresso, sia dall'uscita delle due porte. I flip-flop, che sono chiamati anche bistabili, costituiscono gli elementi fondamentali per la costruzione dei circuiti di memoria.

Floppy disk Dischetto flessibile. Un supporto di memoria di massa costituito da un dischetto di mylar rivestito di materiale magnetico, su cui possono essere immagazzinate informazioni (con un principio analogo a quello su cui si basa la registrazione fonografica). Esistono floppy disk di formati diversi: i piú comuni sui personal computer sono quelli del diametro di 5 pollici e 1/4; i dischetti del diametro di 8 pollici sono usati piú frequentemente su macchine di maggiori dimensioni. Si vanno ora diffondendo anche dischetti di diametro ancora minore, di 3 pollici (chiamati anche microfloppy).

Hard copy Letteralmente, "copia dura", cioè copia permanente dell'output di un calcolatore. Indica, normalmente, uno stampato, in contrapposizione alla "copia non permanente" o "soft copy" fornita da un monitor o dal video del televisore.

Hardware Letteralmente significa "ferramenta" e indica in generale i componenti fisici di un calcolatore: circuiti, cavi, tasti, manopole e via dicendo. È usato in contrapposizione a "software" (→).

Hertz Unità di misura della frequenza equivalente a un ciclo al secondo: è la frequenza di un fenomeno periodico il cui periodo è pari

a 1 secondo. Una corda in vibrazione, per esempio, produce un suono di 400 hertz se compie 400 oscillazioni complete in un secondo. Questa unità di misura prende il nome da Heinrich Hertz, scienziato tedesco che contribuí alla scoperta della natura delle onde elettromagnetiche. Il simbolo relativo è Hz; i multipli piú usati sono il kilohertz (KHz, 1000 hertz), il megahertz (MHz, un milione di hertz) e il gigahertz (GHz, un miliardo di hertz).

IC Sigla di Integrated Circuit, circuito integrato.

Indirizzo (*Address*) Una posizione all'interno della memoria di un calcolatore: la posizione in cui deve essere reperita un'informazione, o in cui l'informazione deve essere scritta. Nei linguaggi di alto livello il programmatore non deve normalmente preoccuparsi degli indirizzi effettivi delle informazioni che vuole manipolare, mentre è indispensabile specificare gli indirizzi nella programmazione in linguaggi di basso livello. L'indirizzamento, cioè il modo in cui è dato l'indirizzo di un'informazione, può avvenire in vari modi: il piú semplice è l'indirizzamento diretto (viene data l'esatta locazione di memoria), ma in genere sono possibili anche altre forme, per esempio l'indirizzamento indiretto (anziché dare direttamente l'indirizzo dell'informazione, si dà l'indirizzo di una posizione di memoria in cui è conservato l'indirizzo dell'informazione) o l'indirizzamento relativo (si dà un valore che va sommato al contenuto di un dato registro per ottenere l'indirizzo effettivo).

Input Ingresso, genericamente inserimento di informazioni.

Input, dispositivo di (*Input device*) Un dispositivo che consente l'ingresso o l'inserimento di informazioni in un computer o in una qualunque apparecchiatura di un sistema di calcolo. Il dispositivo di input tipico è la tastiera.

Intelligenza artificiale (*Artificial intelligence*) Lo studio e la realizzazione di programmi che consentono a un calcolatore di svolgere compiti che, se svolti da un essere umano, diremmo richiedono l'uso di intelligenza. Il confine tra ciò che richiede e ciò che non richiede intelligenza è molto sfumato, ma le ricerche sull'intelligenza artificiale possono essere viste anche come un tentativo scientifico di appurare effettivamente il significato del termine "intelligenza". I campi di studio tipici sono la produzione del linguaggio naturale, la visione e il riconoscimento di forme, i giochi (in particolare quelli che richiedono strategia).

Interfaccia (*Interface*) Un dispositivo interposto fra due apparecchiature, due macchine o due dispositivi diversi, per renderne possibile la comunicazione, adeguando i segnali dell'uno a quelli dell'altro. Il termine è usato anche in senso lato e metaforico.

Interprete (*Interpreter*) Un programma "traduttore" per linguaggi di alto livello: prende una riga di programma alla volta e sovrintende immediatamente alla sua traduzione in codice di macchina e alla sua esecuzione. Linguaggi tradotti da un interprete sono detti "interpretati", in contrapposizione ai linguaggi "compilati" (tradotti da un compilatore). Linguaggi interpretati sono tipicamente il BASIC, l'APL, il LISP, il PROLOG: tuttavia un linguaggio interpretato può essere anche compilato (esistono, per esempio, anche compilatori per il BASIC). L'interprete permette una forma veloce di interazione fra l'uomo e la macchina, in fase di realizzazione dei programmi, ma in fase esecutiva è molto piú lento di un compilatore, perché procede esaminando una sola riga di programma alla volta; se la medesima istruzione si presenta decine o centinaia di volte nel corso di un programma, in righe diverse, deve venir tradotta dall'interprete ogni volta come se fosse un'istruzione mai incontrata.

IR Instruction Register, registro delle istruzioni. È uno dei registri dell'unità centrale di elaborazione (CPU), nel quale viene depositata l'istruzione che deve essere eseguita a quel momento dalla CPU stessa. Può essere (a seconda della macchine) a 4, 8, 12, 16 o 32 bit.

Iterazione → Struttura di controllo.

Joystick Un dispositivo di input che ha la forma di una piccola leva o di una barra, montata su una apposita base. La leva può essere spostata in tutte le direzioni e il suo movimento viene trasformato in impulsi elettrici, interpretati dall'unità di elaborazione come comandi di movimento del cursore o di qualunque altro simbolo predefinito sullo schermo. Il joystick è ampiamente usato come dispositivo di controllo nei giochi (può essere munito anche di un pulsante per le operazioni di "fuoco") per i piccoli calcolatori, ma è usato anche, in campo professionale, come dispositivo di input, al posto della tastiera, per esempio, per programmi di grafica e di progettazione assistita dal calcolatore.

K Prefisso che indica "mille" o, piú precisamente, nel campo dei calcolatori, 1024 (2 alla decima potenza). È usato normalmente per i bit e i byte: 1 Kbit (un chilobit) = 1024 bit. Spesso è usato da solo, per indicare 1024 byte. Nelle descrizioni dei calcolatori, quando si leggono espressioni come "memoria RAM da 128 K" si sottintende sempre che si tratti di Kbyte.

Linker Un programma che prende file di programma in linguaggio macchina prodotti

da compilatori o da assemblatori e li collega ("link" in inglese significa collegare) in un unico file di programma eseguibile.

LISP Acronimo di LISt Processing, indica un linguaggio di programmazione di alto livello, formulato verso la fine degli anni Cinquanta da John McCarthy, orientato in particolar modo all'elaborazione simbolica. È il linguaggio tipico degli studi sull'intelligenza artificiale. Linguaggio tipicamente interpretato, è anche un linguaggio "funzionale": tutte le costruzioni del LISP assumono la forma di funzioni. LISP è anche un linguaggio fortemente ricorsivo.

LOGO Un linguaggio di programmazione di alto livello, formulato da Seymour Papert e collaboratori presso il laboratorio di intelligenza artificiale del Massachusetts Institute of Technology. Derivato dal LISP, è un linguaggio fortemente orientato alla manipolazione simbolica e alla grafica, particolarmente adatto in ambito educativo (Papert è stato influenzato dalle teorie psicologiche e pedagogiche di Jean Piaget), tanto da essersi guadagnato fama di un linguaggio per i bambini. È, tuttavia, un linguaggio ricco e potente, ma facile da utilizzare.

M Prefisso che indica "un milione", o, piú precisamente, nel campo dei calcolatori, 1.024.000! Mbyte (un megabyte) = 1.024.000 byte. Spesso è usato anche da solo, in quest'ultimo significato di "megabyte".

Macchina di Turing (*Turing machine*) Il modello piú astratto e piú potente di macchina calcolatrice, formulato nel 1936 dal matematico inglese Alan M. Turing per dare una definizione rigorosa dei concetti di algoritmo e procedura effettiva. Una macchina di Turing (che, evidentemente, è una macchina ideale), è costituita da un nastro infinito diviso in celle, su cui possono essere scritti simboli tratti da un insieme finito, e da una testina di lettura/scrittura, che può muoversi lungo le due direzioni del nastro di una sola cella alla volta. L'unità di elaborazione di una macchina di Turing è un automa finito, che può assumere un numero finito di stati interni e fornire un'uscita ben determinata (lettura, scrittura, cancellazione di simboli) in funzione dell'ingresso e dello stato in cui si trova.

Mainframe Un calcolatore di grandi dimensioni, dal costo elevato e dalle elevate prestazioni. Il termine è usato per distinguere la fascia piú alta degli elaboratori commerciali, in contrapposizione a "minicomputer" e "microcomputer". Il vocabolo significa letteralmente "armadio principale" e veniva usato originariamente per indicare i grandi telai e i grandi armadi che contenevano le migliaia di valvole termoioniche utilizzate dai primi calcolatori elettronici.

MAR Memory Address Register, registro dell'indirizzo di memoria. È uno dei registri dell'unità centrale di elaborazione (CPU), nel quale viene depositato l'indirizzo della locazione di memoria prescelta, a quel dato momento, per una operazione di lettura oppure di scrittura.

Margherita (*Daisywheel*) Nelle stampanti, un dispositivo di stampa di forma circolare su cui, come i petali di una margherita, sono innestate le asticciole che portano incisi i caratteri. Si tratta di un meccanismo "a impatto" che produce risultati di qualità elevata. Le stampanti a margherita sono usate normalmente per documenti e corrispondenza, là dove l'eleganza e la leggibilità del risultato sono fattori prevalenti rispetto alla velocità e all'economia di stampa.

Megabyte Unità di misura, multiplo di byte: un megabyte è pari a un milione di byte.

Memoria (*Memory, Storage*) Dispositivo di immagazzinamento interno o esterno al calcolatore. La memoria principale, interna alla macchina, è veloce e costosa (di conseguenza in genere abbastanza piccola); quella secondaria o di massa, esterna, è piú lenta e meno costosa (e in genere molto piú ampia).

Menú In un programma, la presentazione sullo schermo di una serie di possibilità di scelta per l'utente. È uno strumento di semplificazione dell'interazione fra utente e calcolatore: l'utente non deve scrivere istruzioni complesse per indicare al programma che cosa desidera, ma è il programma stesso che offre un elenco esaustivo di tutte le possibilità per la situazione data. L'utente può effettuare la sua scelta premendo un solo tasto (un numero o una lettera) corrispondente alla possibilità desiderata.

Microprocessore (*Microprocessor*) Una unità di elaborazione completa, integrata su un unico chip di silicio, anziché essere ottenuta dal collegamento di piú componenti discreti. Il microprocessore ha reso possibile la nascita dei microcomputer, e dei personal computer in particolare: ne costituisce il "cuore", l'unità di elaborazione.

Minifloppy Un floppy disk del diametro di 5 pollici e 1/4.

Modem Acronimo di MOdulatore/DEModulatore. Indica un'apparecchiatura per la trasmissione su una linea telefonica e, nell'altro senso, per la conversione di segnali analogici in arrivo su una linea telefonica in forma digitale, adatta per l'ingresso a un calcolatore. Un modem in senso stretto effettua la conversione direttamente in segnali elettrici; un accoppiatore acustico opera invece una conversione in segnali acustici, che vengono poi trasferiti alla linea attraverso il microtelefono come i normali segnali vocali.

Monitor 1 Un visualizzatore a tubo a raggi

catodici, essenzialmente simile a un televisore, ma privo dell'apparato di ricezione dei programmi televisivi e specificamente studiato per la visualizzazione dell'uscita di un computer. Un monitor può essere monocromatico (caratteri bianchi, ocra o verdi su fondo nero) o a colori.

Monitor 2 Software o firmware che funge da controllore, supervisore e verificatore dell'attività generale di un sistema.

MOS Metal-Oxide-Semiconductor, metallo-ossido-semiconduttore. Si riferisce alla configurazione a tre strati usata nella fabbricazione dei transistor a effetto di campo (→Transistor). La tecnologia MOS presenta bassissima dissipazione di potenza e rende possibile la realizzazione di circuiti ad alta densità di componenti senza creare problemi di surriscaldamento.

Mouse Un dispositivo di ingresso per una unità di elaborazione, che ha esternamente la forma di una piccola scatola con uno o piú pulsanti. Il mouse governa il cursore: spostando il mouse sul piano del tavolo, il cursore si sposta nella stessa direzione, in misura proporzionale. Il metodo è molto utile in abbinamento a programmi che fanno grande uso di menú: con il mouse si porta il cursore sull'opzione desiderata, e con uno dei pulsanti (che svolge una funzione analoga a quella del tasto RETURN) si comunica alla macchina che quella è la scelta effettuata. Il mouse fa parte dei dispositivi studiati per rendere piú amichevole l'interfaccia fra l'utente e la macchina, ed è molto usato in calcolatori orientati alla grafica in tutte le loro operazioni, come è ad esempio il Macintosh della Apple.

MS-DOS Sigla di Microsoft Disk Operating System, sistema operativo sviluppato dalla società americana Microsoft per microcomputer a 16 bit, in particolare per l'IBM Personal Computer e per i suoi derivati.

Multielaborazione (*Multiprocessing*) Si dice dell'attività di un sistema di calcolo comprendente piú unità di elaborazione che lavorano in parallelo sotto la supervisione di un unico sistema operativo, condividendo unità di memoria e dispositivi di ingresso e uscita, e con organi di comunicazione che consentono il passaggio di dati da ciascuna unità di elaborazione alle altre. I vantaggi della multielaborazione sono la maggior potenza complessiva del sistema, che pertanto può eseguire compiti piú complessi, e la sua maggiore disponibilità (un guasto a una delle unità di elaborazione infatti non blocca automaticamente tutto il sistema).

Multiplatore (*Multiplexer*) Apparecchiatura che realizza la funzione di multiplazione. La funzione è simile a quella di concentrazione ma un multiplatore (a differenza di un concentratore) non è programmabile.

Multiplazione (*Multiplexing*) Realizzazione di piú collegamenti logici mediante un unico collegamento fisico.

NAND Uno dei circuiti logici fondamentali di un sistema di elaborazione: equivale a un NOT AND, e quindi dà in uscita valore vero (1) se almeno uno dei due valori in ingresso è falso (0).

Nano Prefisso che, nel sistema metrico decimale, indica un miliardesimo: un nanosecondo è pari a un miliardesimo di secondo.

NOR Uno dei circuiti logici fondamentali di un sistema di elaborazione: equivale a un NOT OR, che dà quindi in uscita valore vero (1) se e solamente se ambedue gli ingressi sono falsi (0).

NOT Un circuito fondamentale di un sistema di elaborazione e un operatore booleano fondamentale. NOT in inglese significa "non". Il circuito accetta in ingresso un segnale logico (0 o 1) e dà in uscita il segnale opposto: realizza cosí il comportamento dell'operatore logico, che dà il vero (1) se applicato a un operando falso (0) e viceversa.

Nuclei, memoria a (*Core memory*) Un tipo di memoria principale largamente usato nei computer fino agli anni Settanta. I nuclei erano anelli di ferrite, del diametro di circa un millimetro e sono stati soppiantati dai dispositivi microelettronici. Il termine inglese "core memory" è usato ancora oggi per indicare genericamente la memoria principale di un computer (anche se è realizzata con tecnologie diverse).

OCR (*Optical Character Recognition*) → Riconoscimento ottico di caratteri.

OR Uno dei circuiti fondamentali di un sistema di elaborazione e uno degli operatori booleani fondamentali. OR in inglese significa "o" e il circuito e l'operatore si comportano come la congiunzione: danno come risultato il vero (1) se almeno uno dei due segnali (uno dei due operandi) è vero (1); in caso contrario danno come risultato il falso (0).

Ospite (*Host*) Nell'elaborazione distribuita, un calcolatore che fornisce servizi o svolge funzioni di controllo nei confronti degli utilizzatori o dei calcolatori satellite, dei terminali e delle altre apparecchiature ausiliarie. Un calcolatore ospite in genere è considerato autosufficiente e non richiede supervisione da parte di altri calcolatori.

Ottale, sistema (*Octal system*) Sistema di numerazione posizionale a base 8, che usa le cifre 0-7 del sistema decimale. In un numero di piú cifre, ogni posizione rappresenta una diversa potenza di 8, come nel sistema decimale ogni posizione rappresenta una potenza di 10.

Output Genericamente, l'uscita da un si-

stema di elaborazione o da una sua componente: i risultati di un'elaborazione, i segnali inviati a un dispositivo periferico,...

Output, dispositivo di (*Output device*) Un dispositivo che consente l'uscita di informazioni dall'unità di elaborazione di un computer, per il loro trasferimento ad altre apparecchiature o per la comunicazione all'utente. Tipici dispositivi di output sono il video, la stampante, il plotter.

PASCAL Linguaggio di programmazione di alto livello sviluppato nel 1969 da Niklaus Wirth della Eidgenossische Technische Hochschule di Zurigo, pensato come linguaggio per l'insegnamento della programmazione come attività sistematica. In seguito ha trovato larga applicazione anche al di fuori dell'insegnamento, in particolare sui personal computer, grazie alla realizzazione di compilatori che richiedono capacità di memoria relativamente piccole. È un linguaggio ricco ed elegante, orientato alla programmazione strutturata, che permette la stesura di programmi ricorsivi.

PC Sigla di Program Counter o contatore di programma. È uno dei registri dell'unità centrale di elaborazione (CPU), nel quale sono depositati gli indirizzi necessari per il controllo della successione delle istruzioni di un programma. Contiene normalmente l'indirizzo della locazione di memoria da cui dovrà essere caricata la successiva istruzione. Durante gli interrupt il PC conserva l'indirizzo dell'istruzione.

PEEK Una istruzione del BASIC e di alcuni altri linguaggi di programmazione, che consente all'utente di conoscere i contenuti di una specifica locazione di memoria della macchina.

Periferica (*Peripheral device*) Genericamente qualunque dispositivo di input/output. Una stampante è una tipica periferica.

PILOT Un linguaggio di programmazione di alto livello, di facile apprendimento, pensato per utenti non esperti, in particolare nell'ambito delle applicazioni scolastiche o attinenti all'istruzione in generale. È fra i piú diffusi "linguaggi autore", sviluppati per la preparazione di lezioni da parte dell'insegnante, che può realizzare le proprie applicazioni senza doversi cimentare con la programmazione in un linguaggio procedurale di tipo algoritmico.

Pixel Abbreviazione di "picture cell", indica l'elemento minimo di un'immagine, in particolare su schermo. Le dimensioni del pixel dipendono dalle capacità grafiche della macchina e dalla capacità di risoluzione del video utilizzato.

PL/I Sigla di Programming Language/One. Linguaggio di programmazione di alto livello, sviluppato negli anni 1963-64 da un comitato fondato dalla IBM. Si proponeva come linguaggio universale, adatto sia per la programmazione scientifica, sia per quella commerciale, sia ancora per la programmazione di sistemi. Ha preso caratteristiche dai tre maggiori linguaggi di programmazione che l'hanno preceduto, e cioè FORTRAN, ALGOL e COBOL, ma ha goduto di un successo limitato, proprio perché l'ambizione all'universalità lo ha reso piú pesante e meno efficiente degli altri linguaggi dall'ambito d'azione meno ampio. Ha avuto diffusione soprattutto grazie al coinvolgimento diretto della IBM nella sua definizione.

Plotter Un dispositivo di output per un sistema di elaborazione, che permette di ottenere disegni su carta o su altri supporti (cartone, plastica, ecc.). I plotter a penna disegnano tramite il movimento relativo di una o piú penne su supporto fisso o mobile (plotter a tavola piana, plotter a rullo, plotter a tamburo). Nei plotter elettrostatici la carta passa sotto una testina (una serie di aghi) e viene sensibilizzata selettivamente dagli aghi elettricamente carichi; viene poi spruzzato un liquido contenente particelle di carbone in sospensione, che vengono attirate dai punti sensibilizzati e, infine, si ha un processo di essiccazione ed eliminazione delle particelle in eccesso. I plotter elettrostatici non hanno parti meccaniche in movimento e risultano pertanto piú veloci.

POKE Una istruzione tipica del BASIC e di alcuni altri linguaggi di programmazione, che effettua l'operazione opposta rispetto all'istruzione PEEK: consente, cioè, di inserire uno specifico valore direttamente in una specifica locazione di memoria della macchina.

Porta logica (*Logic gate*) Qualsiasi circuito che realizzi una funzione logica elementare: NOT, AND, OR, NAND, NOR. Non è importante che il circuito sia di tipo elettronico: sono state realizzate, per esempio, anche porte logiche di tipo ottico (il cui funzionamento, cioè, è basato sullo sfruttamento di meccanismi luminosi). Nei circuiti elettronici, i valori logici "vero" (1) e "falso" (0) sono rappresentati da segnali, in ingresso e in uscita, caratterizzati da due valori distinti di tensione: il livello di tensione elevato rappresenta normalmente il valore 1, il livello basso rappresenta invece il valore 0.

Programma applicativo (*Application program*) Un programma che svolge una funzione direttamente pertinente alle attività dell'utente: un programma di word processing è un programma applicativo, come un programma per il calcolo dell'IVA o per la gestione del proprio conto bancario. Il termine è in opposizione a programma (o software) di sistema: un programma utile o essenziale per il funzionamento del calcolatore, ma non inteso

direttamente allo svolgimento di un'attività per l'utente. Software di sistema è un sistema operativo, un interprete, un compilatore.

PROLOG Acronimo di PROgramming in LOGic, indica un linguaggio di programmazione di alto livello, utilizzato soprattutto per applicazioni di intelligenza artificiale. Formulato agli inizi degli anni Sessanta da Alain Colmerauer dell'Università di Marsiglia, è un linguaggio dichiarativo, anziché imperativo come i linguaggi piú diffusi (BASIC, PASCAL, FORTRAN, COBOL e simili). Le frasi del linguaggio, cioè, non hanno la forma di istruzioni che il computer deve eseguire, ma di dichiarazioni di fatti che la macchina deve memorizzare, e su cui poi potrà fornire informazioni, se opportunamente interrogata secondo le modalità previste dal linguaggio stesso. Linguaggio tipicamente interpretato e per sua stessa natura particolarmente adatto per la costruzione e l'interrogazione di grandi basi di dati "intelligenti" e la costruzione di sistemi esperti, il PROLOG è stato scelto come linguaggio fondamentale nel progetto giapponese di elaboratori della quinta generazione.

PROM Sigla di Programmable Read Only Memory, memoria programmabile a sola lettura. Indica un tipo di memoria ROM, a sola lettura, che può essere programmata dall'utente, ma una sola volta. Il contenuto della memoria, inalterabile una volta scritto, è definito dall'utente anziché dal produttore all'origine. Esistono anche PROM che possono essere cancellate e riprogrammate (➔EAROM).

Pronto (*Prompt*) Un segnale che compare sullo schermo del computer, rappresentato da un quadratino lampeggiante, da due punti, dalla scritta READY o da altro simbolo apposito, e che indica la disponibilità dell'apparecchiatura ad accettare in ingresso informazioni o istruzioni. Spesso si usa l'equivalente denominazione inglese "prompt".

Protocollo (*Protocol*) Un insieme di regole per la trasmissione di dati: ambedue i calcolatori che si trovano alle estremità di una linea di trasmissione debbono utilizzare lo stesso protocollo per poter controllare il flusso delle informazioni.

QWERTY I primi sei tasti da sinistra, sulla prima fila superiore di tasti, in una tipica tastiera americana da macchina per scrivere o da computer. "Tastiera QWERTY" significa una tastiera che segua la stessa impostazione di una tastiera americana standard: rispetto a quella tipica italiana, per esempio, sono invertite le posizioni di Z e W, la M sta dove sulle macchine per scrivere italiane si trova la virgola, e via dicendo.

RAM Random Access Memory, memoria ad accesso casuale. Un tipo di memoria principale in cui è possibile non solo leggere, ma anche registrare informazioni. La memoria RAM è la memoria di lavoro dell'utente, quella in cui vengono caricati i programmi dai dispositivi di memoria di massa e in cui vengono temporaneamente archiviati i risultati delle elaborazioni. È una memoria normalmente volatile: quando si toglie l'alimentazione al computer, i suoi contenuti vengono cancellati. (Fanno eccezione le memorie RAM dei computer portatili, che conservano le informazioni grazie ad una propria alimentazione separata e continua.) Le dimensioni della memoria RAM determinano quindi la complessità dei programmi che possono girare su un computer e le quantità di dati che possono essere disponibili immediatamente, internamente alla macchina.

Registro (*Register*) Un piccolo gruppo di celle di memoria all'interno dell'unità centrale di elaborazione, per conservare informazioni a cui l'unità di elaborazione deve poter accedere molto velocemente. Qualunque unità centrale di elaborazione possiede numerosi registri, alcuni dei quali hanno compiti prestabiliti (il registro delle istruzioni, che conserva l'istruzione che deve essere eseguita, il contatore di programma, che conserva l'indirizzo dell'istruzione da eseguire, ecc.), mentre altri sono a disposizione del programmatore, che può disporne in funzione delle necessità dei singoli programmi.

Riconoscimento ottico di caratteri (*Optical Character Recognition, OCR*) Lettura da parte di una macchina di caratteri dattiloscritti e manoscritti. Esistono periferiche per computer in grado di riconoscere manoscritti con calligrafia molto precisa e regolare: sono tuttavia ancora dispositivi complessi, costosi e poco duttili. Il riconoscimento ottico dell normale calligrafia corsiva è ancora un traguardo lontano.

ROM Read Only Memory, memoria a sola lettura. Un tipo di memoria principale su cui sono registrate informazioni in forma permanente dal costruttore, e su cui l'utente non può "scrivere", ma da cui può solamente recuperare quelle informazioni. Su ROM sono registrati i programmi di base del computer, quelli che ne consentono il "lancio" all'accensione, ovvero la capacità di accettare input dall'esterno.

RS-232 Un tipo di interfaccia standard per trasmissioni di dati tra apparecchiature di elaborazione. È un'interfaccia seriale (la trasmissione dei codici avviene cosí sequenzialmente, bit dopo bit) sviluppata dalla Electronic Industries Association (EIA) con la Bell System, produttori indipendenti di modem e società produttrici di calcolatori. Lo standard specifica un connettore a 25 piedini e i segnali necessari per l'interfacciamento. È l'inter-

faccia piú diffusa nel mondo dei personal computer, per le comunicazioni in via diretta o tramite modem: molte macchine ne sono provviste di serie, per le altre è in genere disponibile come accessorio.

Scheda perforata (*Punched card*) Una scheda speciale in cartoncino in cui vengono praticati dei fori, per il controllo automatico di apparecchiature. Le schede perforate furono introdotte da Jacquard nell'Ottocento per il controllo dei telai da tessitura (i fori consentivano l'abbassamento degli aghi), poi furono adottate da Hollerith nelle sue macchine elaboratrici (progettate per l'elaborazione dei dati del censimento della popolazione degli Stati Uniti del 1900). A lungo le schede perforate hanno rappresentato il mezzo piú economico per l'inserimento di dati nei calcolatori.

Scrolling Scorrimento del testo sullo schermo. Lo schermo di una unità video può essere pensato come una sorta di "finestra" su un foglio molto lungo: lo scrolling equivale allo spostamento delle finestra lungo il foglio per visualizzare parti diverse. Nelle fasi di inserimento di un testo o di un programma lo scrolling è automatico: quando tutte le righe visualizzabili sullo schermo sono occupate, automaticamente la prima scorre fuori dal campo di vista per far posto verso il basso a una nuova riga.

Sequenza → Struttura di controllo.

Set di caratteri (*Character set*) L'insieme dei caratteri disponibili su una determinata apparecchiatura: il set di caratteri di una stampante è l'insieme dei caratteri che può stampare.

Silicio (*Silicon*) Elemento chimico non metallico: è l'elemento piú abbondante sulla Terra, dopo l'ossigeno. Viene utilizzato per la realizzazione di componenti elettronici, celle solari, ecc.

Simulazione (*Simulation*) L'uso di modelli di sistemi dinamici per lo studio sperimentale dei sistemi reali. In genere si intende con il termine "simulazione" che tali modelli siano realizzati al computer, perché questa è la tecnica piú efficace e flessibile. Nella simulazione si studia il comportamento di un sistema in condizioni particolari, per esempio in tutti i casi in cui l'esperimento sul sistema reale non è realizzabile o lo sarebbe solamente con rischi elevatissimi. La simulazione al computer, per esempio, permette a un giovane medico di effettuare interventi chirurgici come se stesse operando su un paziente, ma senza che i suoi errori possano avere conseguenze disastrose; un simulatore di volo permette l'addestramento di un pilota senza mettere a repentaglio la sicurezza di persone e cose.

Sistema esperto (*Expert system*) Un sistema di calcolo programmato in modo da comportarsi come farebbe un esperto, se interrogato su problemi pertinenti al suo settore di competenza. Come un esperto, un sistema di questo tipo possiede una base di conoscenze, regole deduttive rigorose e regole euristiche non rigorose e imprecise, di tipo "intuitivo" ma corroborate dall'esperienza, e fornisce risposte alternative possibili, quantificando la probabilità relativa di ciascuna. Cosí, per esempio, un sistema esperto per la geologia, davanti ai dati raccolti relativamente a una data zona, potrebbe dire che ci sono x probabilità per cento di presenza di un certo tipo di giacimento, y probabilità per cento di presenza di un altro tipo di giacimento, e via dicendo.

SNOBOL 4 Un linguaggio di programmazione di alto livello sviluppato nel corso degli anni Sessanta ai Bell Laboratories, orientato alla manipolazione di stringhe di caratteri. Non possiede strutture a blocchi né dichiarazioni di variabili, ma dispone di molti strumenti per il riconoscimento, la generazione, la cancellazione e la sostituzione di stringhe e sottostringhe.

Software Termine generico, coniato sul calco di "hardware" ("hard" significa "duro", "soft" significa "molle, morbido") per indicare i programmi di un calcolatore (indipendentemente dal suo supporto fisico). In italiano è stata tentata l'introduzione del termine "logicale" (sul modello del francese "logiciel"), che però non ha avuto fortuna ed è utilizzato da pochissimi.

Stampante (*Printer*) Una periferica di uscita di un sistema di elaborazione, che consente la visualizzazione su carta. Esistono molti tipi di stampanti, che si raggrùppano solitamente in due categorie: a impatto e non a impatto. Nelle stampanti a impatto il meccanismo di scrittura entra in contatto con la superficie del foglio, con un principio analogo a quello di una macchina per scrivere elettrica o mediante una serie di aghi che vanno a premere il nastro inchiostrato. Nelle stampanti non a impatto il meccanismo non entra in contatto con la superficie del foglio, ma crea il carattere con procedimenti termici (su carta opportunamente trattata chimicamente) o spruzzando inchiostro (stampanti a getto d'inchiostro) o ancora con procedimento simile a quello della riproduzione xerografica.

Stringa (*String*) Calco del termine inglese, che significa letteralmente "fila, successione, corda". Indica una sequenza di caratteri alfanumerici.

Struttura di controllo (*Control structure*) Nei linguaggi di programmazione, i costrutti che definiscono l'andamento del flusso di operazioni della macchina. La struttura piú semplice, e normalmente implicita, è la sequenza: le istruzioni di un programma vengo-

no eseguite una dopo l'altra, nell'ordine in cui sono state scritte. Le altre due strutture di controllo fondamentali permettono di modificare questo flusso sequenziale: l'iterazione permette di ripetere ciclicamente un gruppo di istruzioni (per un numero prefissato di volte, oppure finché vale, o fino a che non si verifica una determinata condizione) e l'alternativa permette di eseguire l'una o l'altra fra due istruzioni (o fra due gruppi di istruzioni) a seconda che valga o meno una data condizione.

Teletext Un servizio di informazione di tipo telematico, unidirezionale (dalla sorgente verso l'utente), inviato come parte del segnale televisivo. L'informazione può essere visualizzata su un normale televisore domestico, dotato di un opportuno decodificatore. L'informazione si presenta in "pagine", selezionabili mediante la pulsantiera di un telecomando, e può essere solo in quantità limitate. Il sistema può servire solamente a trasmettere informazioni di utilità generale come notizie di attualità, notizie metereologiche, quotazioni di borsa, orari di voli aerei, notizie di carattere locale. Il servizio è attivo dal 1984 anche in Italia, fornito con il nome di "Televideo" dalla RAI.

Traccia (*Track*) Un'area di memorizzazione magnetica, di forma circolare, su un disco. Il numero delle tracce presenti su un dischetto può variare, in funzione delle caratteristiche del supporto fisico stesso e del tipo di formato che ciascun calcolatore impone.

Transistor Un componente elettronico che può svolgere funzioni di commutatore o di amplificatore. Il transistor, presente nei piccoli apparecchi radioriceventi e negli amplificatori, è il componente fondamentale dei circuiti integrati, nei quali si trova in forma miniaturizzata. Su un chip di silicio di pochi millimetri di lato possono trovare posto migliaia di transistor.

Trasferimento elettronico di fondi (*Electronic Fund Transfer, EFT*) Trasferimento da un conto bancario a un altro di somme di denaro mediante tecnologie elettroniche. Per esempio, un terminale di computer in un negozio o in un grande magazzino può leggere la carta di credito di un cliente e, collegandosi al computer della banca del cliente e a quello della banca del negozio, ordinare direttamente il trasferimento della somma necessaria per un acquisto, dal conto bancario del cliente al conto del negozio.

UART Sigla di Universal Asynchronous Receiver/Transmitter (ricevitore/trasmettitore asincrono universale). Un dispositivo che trasforma dati paralleli in forma seriale per la trasmissione lungo un'interfaccia seriale, o viceversa. L'UART si rende necessario perché la CPU trasferisce le informazioni in parallelo sul bus dei dati e degli indirizzi, mentre in genere i dispositivi periferici lavorano in modo seriale. L'UART è universale nel senso che può elaborare informazioni per tutti i tipi di dispositivi, ciascuno dei quali richiede una particolare velocità di trasferimento, un particolare tipo di controllo di parità, un particolare metodo per separare un byte di informazione dall'altro. La denominazione di "asincrono" deriva dal fatto che la distribuzione nel tempo degli impulsi di trasferimento è diversa da quella dell'unità centrale.

Unità centrale di elaborazione → CPU.

Unità di controllo (*Control unit*) La parte dell'unità centrale di elaborazione di un computer che presiede al controllo di tutti i trasferimenti di informazione fra unità aritmetica e logica, memoria, unità di ingresso e uscita.

User friendly Espressione gergale che significa letteralmente "amichevole per l'utente". Viene usata genericamente per indicare una apparecchiatura, un programma, o qualunque aspetto di un sistema di calcolo, che risulti facile da apprendere e da maneggiare.

Volatile Si dice di una memoria i cui contenuti si perdono quando viene tolta l'alimentazione. La memoria RAM di un computer è in genere volatile; è non volatile, però, la RAM dei computer portatili, che conserva dati e programmi anche a calcolatore spento. È non volatile la memoria ROM.

Winchester Una tecnologia di costruzione di dischi rigidi per la realizzazione di sistemi di memoria di massa per calcolatori, sviluppata negli anni Settanta nei laboratori della IBM. Un disco Winchester è un disco magnetico rigido in un contenitore sigillato; le operazioni di lettura e scrittura sono effettuate da una testina che non viene mai in contatto con il disco stesso, con possibilità di danneggiamento nulle. Il contenitore sigillato impedisce la penetrazione di agenti contaminanti dall'esterno. Grazie a queste caratteristiche, un disco Winchester può immagazzinare grandi quantità di informazioni (da 5 a 20 milioni di byte e anche più).

Word processing → Elaborazione di testi.

Word processor Una apparecchiatura elettronica digitale specificamente progettata per l'elaborazione di testi. Si tratta, in sostanza, di un calcolatore "dedicato" a questo specifico compito.

XOR Uno dei circuiti logici fondamentali di un sistema di elaborazione: corrisponde a un OR esclusivo, che dà in uscita cioè il valore vero (1) se uno solo dei due ingressi è vero, e dà il valore falso (0) quando i due ingressi sono ambedue falsi o ambedue veri.

Bibliografia

Sui personal computer sono stati scritti e pubblicati moltissimi libri: chi volesse approfondire qualche argomento ha solamente l'imbarazzo della scelta. Ne segnalo qui alcuni, che mi sembrano particolarmente utili o ben fatti.

Chiunque voglia mantenersi aggiornato, o approfondire argomenti molto specifici, è bene tenga d'occhio il catalogo di almeno quattro case editrici, che sono specializzate o hanno almeno un settore specializzato nel campo dell'informatica personale: in ordine alfabetico, Gruppo Editoriale Jackson (Milano), McGraw-Hill italiana (Amburgo), Mondadori (Milano), Muzzio (Padova).

Per restare su libri di carattere generale, si possono vedere:
Bradbeer R., De Bono P., Laurie P., *Capire il computer*, Mondadori, Milano, 1984.
Cavalcoli A., *Scegliere il personal computer*, Mondadori, Milano, 1983.

Per la programmazione e i vari linguaggi:
Abelson, Di Sessa, *La geometria della tartaruga*, Muzzio, Padova, 1985.
Banfi G., *Facile come il Basic*, Mondadori, Milano, 1983.
Campi R., *Programmare è progettare*, Mondadori, Milano (in preparazione).
Chirlian P., *Programmare in C*, Muzzio, Padova, 1985.
Dwyer T., Chritchfield M., *Il Basic e il personal computer*, Muzzio, Padova, 1982.
Fum D., *Programmare in Logo*, Muzzio, Padova, 1985.
Grogono P., *Programmare in Pascal*, Muzzio, Padova, 1983 (seconda edizione in preparazione).
Le Beux P., *Introduzione al Basic*, Gruppo Editoriale Jackson, Milano, 1981.
Patts R.E., *Programmare con Karel il robot*, Mondadori, Milano, 1984.
Reggini H., *Logo: ali per la mente*, Mondadori, Milano, 1984.
Walsh D., *La programmazione in Pascal*, Franco Angeli, Milano, 1982.

Sull'hardware dei computer si possono vedere:
Cripps M., *L'hardware dei computer*, Muzzio, Padova, 1982.
Gilmore C. M., *Introduzione ai microprocessori*, Zanichelli, Bologna, 1983.

Sull'intelligenza artificiale non esiste molto, nel momento in cui scrivo, in lingua italiana, ma molti volumi mi risultano in traduzione. Il piú bello è probabilmente:
Winston P. H., *Artificial Intelligence*, Addison Wesley, 1984 (in traduzione presso il Gruppo Editoriale Jackson).
Piú semplice, e molto adatto come prima introduzione, è:
Raphael B., *The Thinking Computer*, Freeman, 1976 (in traduzione presso Muzzio, Padova).
Già pubblicato in italiano è:
Krutch J., *Esperimenti di intelligenza artificiale*, Muzzio, Padova, 1985.

Per le applicazioni esistono molti volumi specializzati per singoli programmi e singole macchine. Di carattere abbastanza generale sono tuttavia:
Haus G., *Informatica musicale*, Gruppo Editoriale Jackson, Milano, 1985.
Polillo R. e altri, *Il software sulla scrivania*, Mondadori, Milano, 1985.
Pozzoli C., *Scrivere con il computer*, Mondadori, Milano, 1984.
Sala V., *Guida al word processing*, Muzzio, Padova (in preparazione).

Per ciò che riguarda il campo della robotica, una facile introduzione è:
Logsdon T., *Robot*, Supernova Edizioni, Milano, 1985.

Indice analitico

I numeri in corsivo si riferiscono alle didascalie; la sigla "gl" indica che il termine è riportato nel glossario.

Abaco *20*
Accesso, metodo di gl
–, tempo di gl
Accesso casuale gl
– diretto gl
– sequenziale gl
Accoppiatore acustico *75*, 110-112, *110*, gl
Accumulatore gl
A/D converter vedi Convertitore A/D
ADA gl
Address vedi Indirizzo
ALGOL gl
Algoritmo 151-152, 153-154, gl
– di Euclide 152, 153, *154*
ALU (Unità aritmetica e logica) 132-135, *135*, gl
Analizzatore lessicale 148
– sintattico 148
Analogico gl
AND 127, 128, *128*, 129, gl
APL 149
Architettura gl
Archivio di dati 42-49
Arithmetic and Logic Unit vedi ALU
Artificial Intelligence vedi Intelligenza artificiale
ASCII 105, 122, *122*, 127, 128, *146*, gl
Assemblatore 146, gl
– linguaggio di *146*, gl

Babbage, Charles *20*
Banca dati 111-112, 166
Base di dati (Data base) 42-49, gl
– distribuita gl
BASIC 18, 19, 22, 23, 67, *75*, 78, 149, 155-160, *160*, 162, gl
Baud gl
BCD gl
Benchmark gl
Bidirezionale gl
Binario, sistema 118-120, *119*, *120*, 122-126, gl
Bistabile flip-flop vedi Flip-flop
Bit 8, 9, 72
– di parità gl
– di stop gl
Boole, algebra di 126-128, gl
Boole, George 126
Booleana, funzione gl
Bootstrap 19, 77, gl
Buffer gl
Bug gl
Bus 132-133, *135*, gl
– degli indirizzi 132, *135*
– dei dati 132, *135*
– di controllo 132, *135*
Byte 73

C 161, gl
CAD gl
CAI gl
Calcolatore elettronico *20*, *65*, *169*
– configurazioni 112-115
Calcolatrice 12
CAM gl
Carattere gl
Cartridge vedi Cartuccia
Cartuccia *75*, *78*, 79, 83, gl
Cassetta 79-82
Catasta gl
Character set vedi Set di caratteri
Chilobyte 75
Chip *21*, *26*, *72*, *73*, *75*, gl
Ciclo delle istruzioni 136-139
– di macchina 136
Circuito integrale *65*
– logico *135*
– semisommatore 128-130
– sommatore 128-130
Clock 71, 132, gl
C-MOS 78
COBOL 148-149, gl
Codice gl
– ASCII vedi ASCII
Comando gl
Compatibilità 52-53, gl
Compilatore 146-148, gl
Computer 7-13
– graphics 96, gl
Concentratore gl
Contatore delle istruzione (di programma) 135-136
CONTROL 28
Control unit vedi Unità di controllo
Conversione gl
Convertitore A/D 141, gl
Convertitore D/A 141, gl
Core memory vedi Nuclei, memoria a
CPI gl
CPS gl
CPU (Unità centrale di elaborazione) 71-76, gl
CRI gl
Cursore 19, *33*, *34*, 94, gl

D/A converter vedi Convertitore D/A
Daisywheel vedi Margherita
Data base vedi Base di dati
Data Base Management System (DBMS) 50-52, gl
Debugging gl
Decimale, sistema 118
Dedicato gl
DELETE 29
Diagramma di flusso (Flow chart) *13*, *23*, 153, *153*, *154*, gl
Didattica *57*, 61-65
Digitale gl
Disco flessibile (Floppy disk) *75*, *80*, *81*, 83-86, *84*, gl
– magnetico *65*
– ottico *87*
– rigido (Hard disk) *75*, *84*, 86-88, gl
Drive 83, *84*, gl

EAROM gl
EDP gl
EFT vedi Trasferimento elettronico di fondi
Elaborazione di testi (Word processing) 24, 38-40, *61*, gl
ENIAC *9*, gl
ENTER 19
EOM gl
EOT gl
EPROM 139-140, *139*, gl
Esadecimale, sistema 121-122, 137, gl
ESCAPE 29, *33*
Expert system vedi Sistema esperto
EXT gl

File 25, *31*, gl
Filoinformazione *48*
Filtro gl
Finestra *42*, 69, *69*, *159*
Flag 134, *135*
Flip-flop 130-132, *131*, gl
Floppy disk vedi Disco flessibile
Flow chart vedi Diagramma di flusso
Formattamento 84-85
FORTH 149, 161, 170
FORTRAN 149, 150, 161
Full adder 129
Funzione cablata 138-139
- programmata 138-139

Generatore di codice 148
Giochi elettronici *63*, 66
Giustificazione 28
GOSUB 158
GOTO 158
Grafica *46-47*, 58-60, *61*, 94, 95

Half adder 129
HALT 138
Hard copy 100, gl
Hard disk vedi Disco rigido
Hardware 37, 71-115, gl
Hertz gl
Hexadecimal system vedi Esadecimale sistema
Hollerith, Herman *21*
Home computer 12-113
Host vedi Ospite

IC gl
Icone *92*, 94
Indirizzo gl
Input 15, gl
Input, dispositivo di 88-96, gl
INSERIMENTO 29
Integrazione 41-42, *65*
Intelligenza artificiale *169*, 172-173, gl
Interfaccia 71, 110-111, *110*, 140-141, gl
Interprete 146-148, *147*
Invertitore *128*
IR gl
Iterazione vedi Struttura di controllo

Joystick 66, *75*, *94*, 95, gl

K 68, 75, gl
Kemeny, John G. 155
Kutz, Thomas 155

Linguaggio di programmazione 143-163, *146*, *147*, *151*, *163*
- di alto livello 145, 146, *147*, *151*
- di basso livello 145
- di medio livello 146
Linguaggio macchina 143-145, 146, *146*
Linker gl
LISP 149, 161-162, 163, *163*, gl
Logic gate vedi Porta logica
Logical shift 127
LOGO 163, *163*, gl
Loop 152, 153
Lovelace, Ada *21*

M gl
Macchina di Turing gl
- tabulatrice *21*
Mainframe 165, gl
MAR gl
Margherita *104*, *106*, gl
Matita luminosa *75*, *94*, 95-96
Megabyte 86, gl
Memoria 74-88, gl
- a bolle magnetiche *81*, *84*, 88
- ad accesso casuale vedi RAM
- a sola lettura vedi ROM
- di massa 16-17, 77, *78*, 83-88
- di trasporto vedi Buffer
Menú 25, *35*, *69*, 93, gl
Microfloppy *81*, 86
Microfono *75*
Microprocessore 71-76, *75*, 132, 133-136, *133*, *135*, 138, 143, gl
Microprogrammazione 139
MICROPROLOG 163
MIDI vedi Musical Instruments Digital Interface
Minifloppy gl
Modem *75*, 110-112, 141, gl
Modo differito 17
- immediato 17
MODULA-2 161
Monitor *75*, *97*, 99-100
Monitor 1 gl
Monitor 2 gl
MOS gl
Mouse *69*, *75*, *92*, 93-94, 95, gl
MS-DOS gl
MSX 160
Multielaborazione gl
Multiplatore 133, gl
Multiplazione gl
Multiprocessing vedi Multielaborazione
Musica 60
Musical Instruments Digital Interface (MIDI) *53*, 60

NAND 127, 128, *128*, gl
Nano gl
Nastro magnetico *65*, *75*
Nepero *20*
NOR 127, 128, gl
NOT 126, 127, 128, *128*, gl
Nuclei, memoria a gl

OCR vedi Riconoscimento ottico di caratteri
OR 127, 128, *128*, gl
Orologio sincronizzatore vedi Clock
Ospite gl
Ottale, sistema 121-122, gl
Overflow 130, 135
Output 15, gl
- dispositivo di 97-110, gl

Paddle 95
Papert, Seymour 163
Parity bit vedi Bit di parità
PASCAL *159*, 161, gl
Pascal, Blaise *20*
Pascalina *20*
PC gl
PEEK gl
Penna ottica *94*, 95-96
Periferica gl
Personal computer 7-13, 113-114, 166-169
- robot 169-170, *171*
PILOT gl
Pixel 100, gl
PL/I gl
Plotter *75*, 108-110, *109*, *110*, gl
POKE gl

Porta logica gl
Posta elettronica 166
Potenziometro *94*, 95
PRINT 22
Printer vedi Stampante
Programma 12, 17-29, *41*
- applicativo *42*, gl
- compilatore 146-148, *147*
- interprete 146-148, *147*
- traduttore 144-145, 146, *147*
Programmazione 136-139
Programmi di comunicazione 56-58
- integrati 42
PROLOG 162-163, gl
PROM 139-140, gl
Pronto (Prompt) 19, gl
Protocollo gl
Punched card vedi Scheda perforata

QWERTY gl

RAM *75*, 77, 78, 79, 136, 139-140, gl
Random access vedi Accesso casuale
Record 44, 45, 48
Registratore a cassette *78*, 82
Registro 127, 133, 134-135, *135*, gl
REM 158
Rete di calcolatori *75*
Retroazione 19
RETURN 19
Riconoscimento ottico di caratteri gl
Robot 169-172, *171*, *173*
ROM 18, 19, *75*, *77*, 78, *78*, 83, 104, 139-140, gl
RS-232 110-111, 141, 170, gl
RUN 23

Scheda perforata *65*, gl
Scorrimento logico 127
Scrolling gl
Sequencer 60
Sequenza vedi Struttura di controllo
Sequenzialità *78*
Serial access vedi Accesso sequenziale
Set di caratteri gl
- di istruzioni *137*, 143
Silicio gl
Simulazione *59*, 65, gl
Sistema esperto 172-173, gl
SNOBOL4 149, gl
Software 37-69, gl
- integrato 67-69
Sommatore 129
Sovrascrittura 29
Spreadsheet vedi Tabellone elettronico
Stampante *75*, 100-108, *101*, *102*, *105*, *106*, *107*, 140-141, gl
- a getto d'inchiostro 106-107
- a impatto 100-104, *102*, 108
- a matrice di punti 101, 104, *102*, *105*, *106*, 108
- laser *102*, 107-108, *107*
- termica *102*, 105, 106, 108
Stop bit vedi Bit di stop
Storage vedi Memoria
Stringa gl
Struttura di controllo 153, 155, gl
Tabellone elettronico (Spreadsheet) 54-56
Tastiera *75*, *88*, 89-93, *90*
Tavola di verità 124-126
Tavoletta grafica *75*, 95-96
Telaio Jacquard *119*
Teleconferenza 168
Teletext gl
Televisore *18*, *75*
Touch sensitive screen *99*
Traccia gl
Transistor *65*, 117, gl
Trasferimento elettronico di fondi gl
Turing machine vedi Macchina di Turing

UART gl
Unità aritmetica e logica vedi ALU
Unità centrale di elaborazione vedi CPU
Unità di controllo gl
User friendly gl

Valvole termoioniche 117
Video 15, 97-100
Videodisco digitale *81*, *87*, 88
Visicalc 55
Volatile gl

Winchester 87-88, gl
Word processing vedi Elaborazione di testi
Word processor gl
Word wrap 28

XOR 126, 127, 128, *128*, 129, gl

Z-80 76, 137, *137*

Referenze iconografiche

Apple, Milano: 38, 39, 42, 80c/b, 93, 96bs, 103a, 110, 159. – Archivio Mondadori, Milano: 9, 11a, 19, 20, 21, 26bs, 27bs, 51, 53a, 54, 55, 58, 59, 60, 61a, 62, 66, 67, 75, 79s, 85, 92s, 94b, 95b, 97, 99, 102, 103b, 106b, 106d, 107a/cs/b, 109, 114, 115, 118, 135a, 139, 151, 163, 167, 169, 173, 174; Mario De Biasi: 26c/bd, 27a/bd; Mauro Galligani: 171; Nino Leto: 170. – Bit Relations, Milano: 11bs, 69, 81ab. – Bits & Bytes, Milano: 63. – Commodore Italia, Milano: 50, 98, 113as. – Deltaprint, Verona: 78, 79d, 80a, 88, 90, 91, 92d, 94a, 95a, 96a, 113ad. – Dorling Kindersley Ltd., Londra: 84b. – Luis A. Frank: 10b. – Erich Hartmann/Magnum Photos, Milano: 10a. – Honeywell Information System, Milano: 72, 73, 104, 105. – I.B.M., Milano: 11bd, 107cd, 113b. – Gianfranco Motto, Milano: 30, 31, 32, 33, 34, 35. – Nasa: 119. – Olivetti, Ivrea: 43, 96bd, 106s, 112b. – Mario Russo, Milano: 164. – Vittorio Salarolo, Verona: 13, 16, 23, 41, 46, 47, 48, 52-53b, 56, 57, 61b, 74, 81as/c/b, 84a, 101, 120, 128, 129, 130, 131, 133, 134, 135b, 146, 147, 150, 153, 154, 168. – SGS-ATES, Milano: 26a, 27cs. – Lino Simeoni, Verona: 17. – Sinclair, Milano: 112a. – Thomson-CSF, G. Warrin: 87. – Emilio Tremolada, Milano: 6, 14, 36, 70, 116, 142. – Andrea Zani, Milano: 111. –

L'Editore ringrazia la Esacomp s.r.l. di Verona e la VECOMP s.r.l. di Verona per il materiale messo a disposizione per le riprese fotografiche.